

AD-A112 604

SIGNAL TECHNOLOGY INC SANTA BARBARA CA

F/S 17/2

ALGORITHM DEVELOPMENT OF A TELEPHONE SIGNAL CONDITIONER FOR THE--ETC(U)

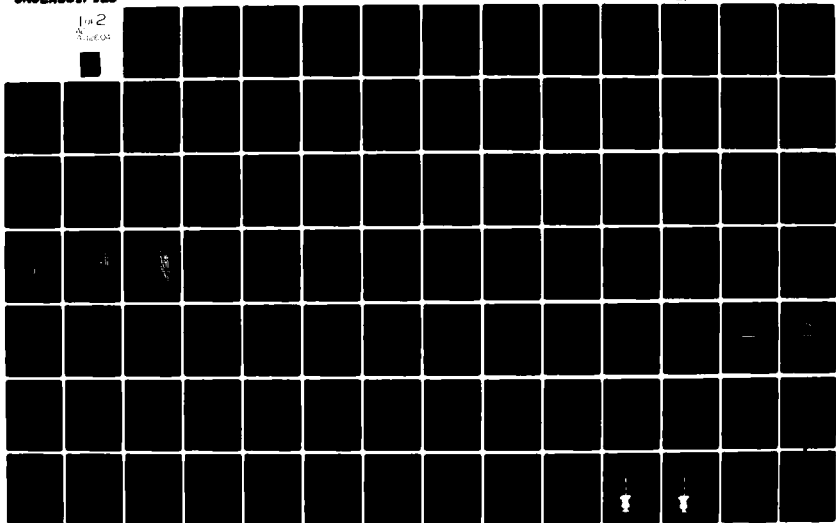
DEC 81 A H GRAY, D MANSOUR

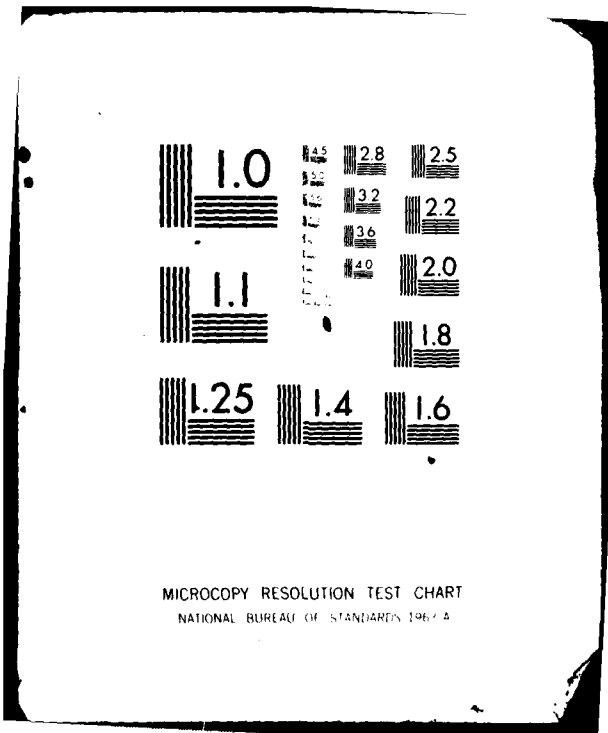
N00016-78-C-0210

ML

UNCLASSIFIED

1 of 2
7/10/04





12

AD A112604

FINAL REPORT
SUMMARIZING CONTRACT NO. N0014-78-C-0214
OVER PERIOD JANUARY 1, 1978 - DECEMBER 31, 1981

ALGORITHM DEVELOPMENT OF A TELEPHONE
SIGNAL CONDITIONER FOR THE
WIDEBAND INTEGRATED NETWORK

A.H. Gray, Jr.
D. Mansour

December 31, 1981

DTIC FILE COPY

Signal Technology, Inc.
15 West De La Guerra
Santa Barbara, CA 93101
805/963-1552

This document has been approved
for public release and its
distribution is unlimited.

DTIC
MAR 5 0 1982

E

82 02 16 028

1 JULY 1976

F223

DEPARTMENT OF DEFENSE FORMS

F-200.1473 DD Form 1473: Report Documentation Page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER FINAL	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Algorithm Development of a Telephone Signal Conditioner for the Wideband Integrated Network.		5. TYPE OF REPORT & PERIOD COVERED FINAL 1/1/78-12/31/81
7. AUTHOR(s) A.H. Gray, Jr. & D.M. Mansour		8. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Signal Technology, Inc. Santa Barbara, CA 93101		10. CONTRACT OR GRANT NUMBER(s) N0014-78-C-0214
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE Dec. 31, 1981
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 121
		14. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) See attached page.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Signal Conditioner, Wideband Integrated Network, Adaptive Filters, Vocoder Speech Transmission		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A complete simulation of interfacing a vocoder to the wideband integrated network is presented, along with theoretical and experimental results.		

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

F-200.1473

ARMED SERVICES PROCUREMENT REGULATION

Table of Contents

1. Introduction	1
2. Test-bed Simulation	11
2.1 Introduction	11
2.2 Interconnection Problems With STN-WIN	12
2.3 The Over-all test-bed Simulation	17
2.3.1 Local Telephone Line Simulation	18
2.3.2 Wideband System Simulation	20
3. Hybrid Simulation	25
3.1 Transhybrid Measurements Under Linear Assumption	27
3.2 Transhybrid response under nonlinear assumption	32
3.2.1 Time-domain Nonlinear Adaptive Filter	33
3.2.2 Measurement Results of a Nonlinear Model	36
4. Echo Cancelling Algorithms	41
4.1 The Gradient Lattice Algorithms	42
4.2 The Unconstrained Frequency Domain LMS Adaptive Filter	45
4.2.1 Introduction	45
4.2.2 The Unconstrained Frequency Domain Adaptive Filter	48
4.2.3 Simulation Results	58
4.2.4 Complexity of the Algorithm	66
4.3 The Double Talker Algorithm	68
5. Improved LPC Analyzer for Large Dynamic Range Signals	70
5.1 Introduction	70
5.2 AGC Interfacing Problems	72
5.2.1 Stability of a System Without an Echo Canceller	73
5.2.2 AGC--Echo Canceller Interaction	77

Accession For	NTIS	SPACI	DTIC TAB	Unannounced	Just in case	Aviation	Dist
	X						
							A

5.3	Pitch and Voicing Algorithm Improvement	80
5.3.1	Modifications	81
5.4	Experimental Results	87
5.5	Conclusion	94
6.	Experimental Results	95
6.1	Introduction	95
6.2	Test-bed Simulation Options	96
6.3	Experimental Results	97
6.3.1	Experiment 1: Echo Effect on Speech Perception as a Function of the Delay Length	97
6.3.2	Experiment 2: Effect of LPC Vocoder in the Full Duplex Network Without An Echo Canceller	99
6.3.3	Experiment 3: Performances of Echo Cancellers Without a Vocoder	100
6.3.4	Experiment 4: Performance of an Echo Canceller with LPC Vocoder in the Loop	102
6.3.5	Experiment 5: Performance of Echo Cancellers with Time-variant Transhybrid Response	106
7.	Summary	113
8.	References	115
9.	Bibliography of Documents Prepared for N0014-78-C-0214	118
10.	Distribution List	120

1. Introduction

Because of the number of practical applications [1] of wideband network with the narrowband Speech Processor (NSP), and wide and easy access of the telephone system, a great deal of interest has been shown in interfacing the telephone line to the wideband network. The effect of this interfacing on the narrowband speech processor, however, must be carefully considered.

The fact that all narrowband speech processors at present are based upon a linear all-pole model of speech production (for voiced speech) is important. If speech is recorded under carefully controlled conditions, i.e., very high signal-to-noise ratio, minimal room reverberation, minimal phase distortion and a high quality microphone, then high quality synthetic speech can be generated with a carefully implemented simulation. However, it has been demonstrated that even relatively small amounts of additive white noise cause the LP coefficients to represent wider bandwidth resonances, resulting in the perception of buzziness in the synthesis. Thus, spectral distortion of the input signal occurs. Two examples:

1. If a tone and a speech signal having roughly equal powers are added, a human listener will perceive the resulting interference as a narrowband whistle. During

analysis and synthesis several of the NSP linear prediction coefficients will partially represent the high amplitude tone, thus reducing the accuracy of the speech representation.

2. If speech is bandpass filtered to 0.3-3.0 kHz, a listener will perceive the speech as "hollow" even though no gross distortion has occurred. Depending on the resistance of the pitch extraction method to the loss of harmonic structure in the spectrum, the synthesis may remain unchanged, or it may become totally unacceptable, due to gross errors in the pitch period tracking.

Figure 1 is a set of block diagrams which illustrates two different examples of interfacing the telephone line to the wideband network. The major difference between these block diagrams is the location of the hybrid, which is used to convert the 2-wire telephone line to 4-wire telephone line. In the configuration of Figure 1 (a), the hybrid and digital processing unit, which includes a low pass filter, automatic gain control, A/D, D/A, analyzer, synthesizer, double talker detector and echo canceller, are at two different locations. This separation makes the problem a little more difficult for the echo canceller. Here we are interested in the configuration of Figure 1(b), where the hybrid and the digital processor are at the same location.

The block diagram of Figure 1(b) can be represented in

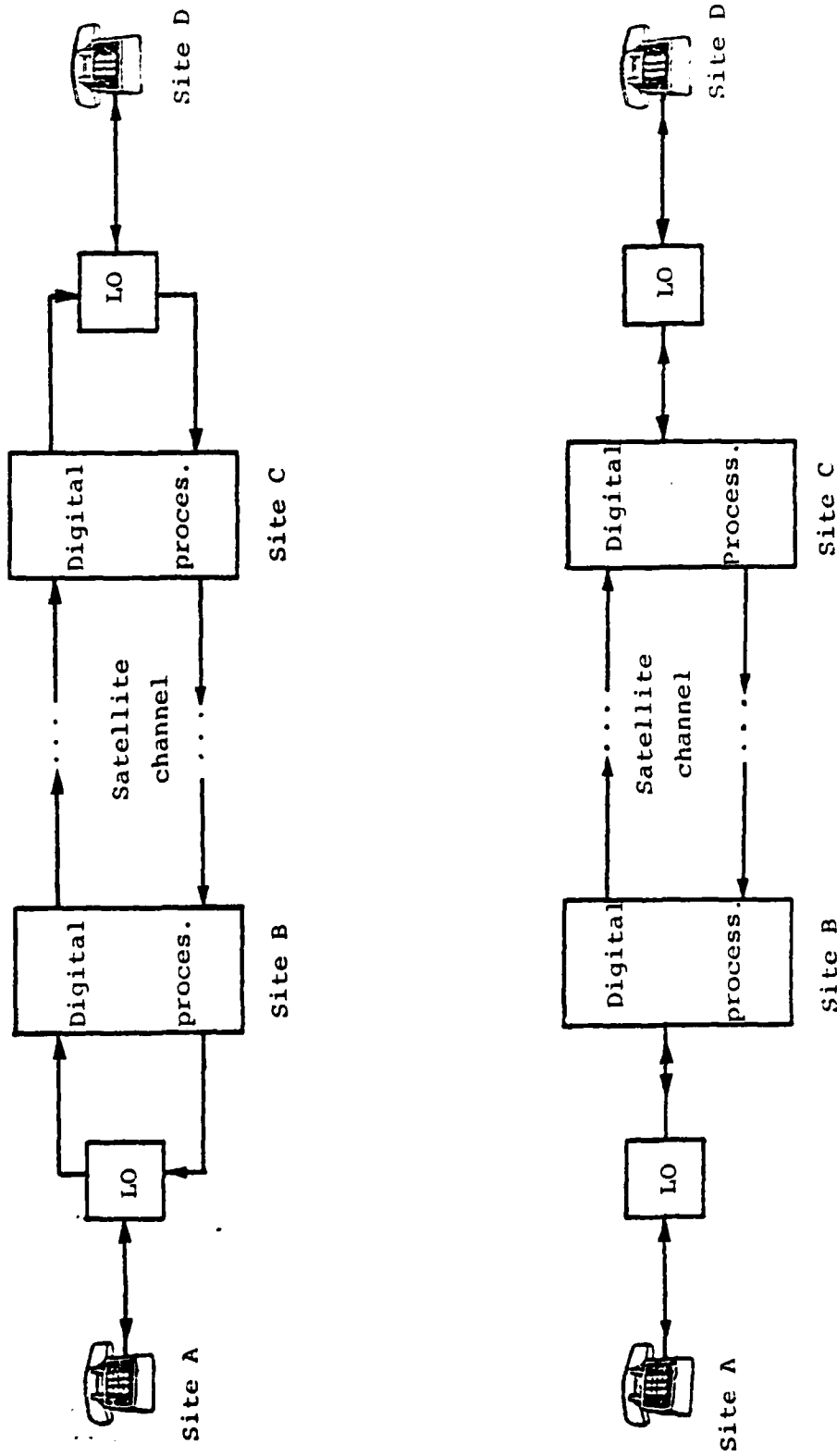


Fig. 1. Block diagrams of typical methods of interfacing the telephone line to the wideband system.

more detail as shown in Figure 2. The user at site A, without any local site equipment except for the telephone, wants to access a user at site D. The sites B and C both have a complete digital speech processing system and also have a leased line channel between sites B and C. The user at site A might be motivated to dial to site d via sites B and C for economic and efficiency considerations or for security considerations. The site A has to access site B via a local or a long distance call depending on the location of site B. This applies to site C and D also. When sites A and D get access to sites B and C respectively, via local calls as shown in Figure 2, then first the user of site A is connected to site B by a 2-wire line through a local office of site A and site B. In the case of long distance calls the site A is connected to the local office of site A by a 2-wire line, then the local office of site A is connected to the site B local office by a "trunk" which is typically a 4-wire circuit. At the site B local office, the 4-wire circuit is converted to a 2-wire circuit which is then connected at site B. In any case, the initial connection at site B is a voice connecting arrangement (VCA). This device mainly protects the telephone line from any customer equipment. Also, it performs supervisory functions such as ringing detection, off hook signaling and automatic seizing of the line. Next, the VCA is connected to a hybrid which converts the 2-wire to the 4-wire

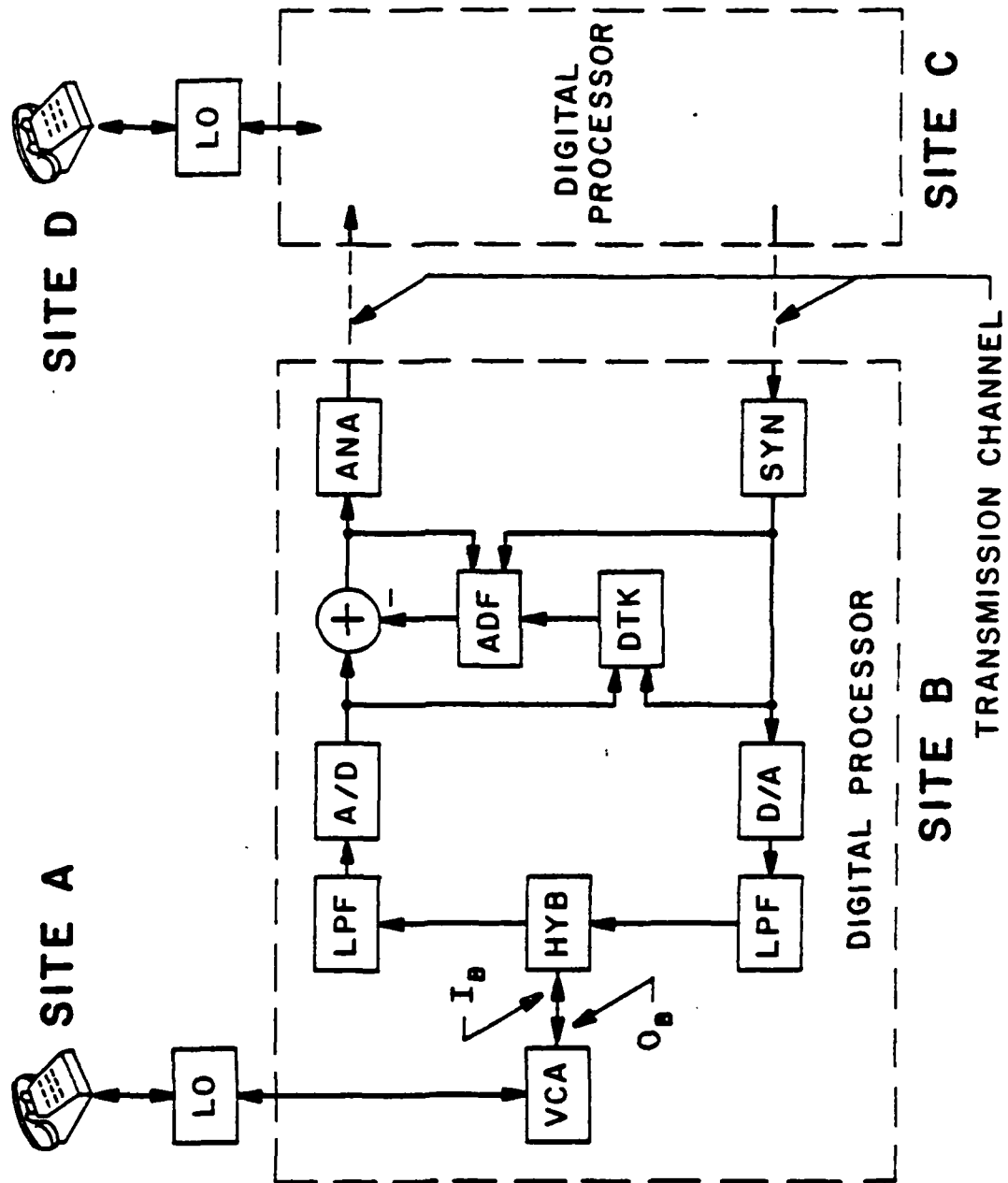


Fig. 2. Block Diagram of Telephone Line Interfacing to the Wideband System.

connection, which in turn is connected to the digital processing system. This system contains a low pass filter for anti-aliasing, automatic gain control to compensate for telephone line loss and maximize the signal to quantizing noise ratio, A/D and D/A, analyzer, synthesizer, double talker detector and echo canceller.

The 2-wire to 4-wire conversion typically results in some of the analog output of the synthesizer being fed back to the analog input of the analyzer, with the amount depending on how well the hybrid is balanced. This leakage signal results in an echo. At site C the process is duplicated.

There are four major sources of distortions which affect the quality of the narrowband speech processor:

- 1) The spectral distortion due to the carbon button microphones, inductive loadings on 2-wire lines and impedance mismatch throughout the system;
- 2) the signal loss, hence signal to noise ratio decreases, which is a function of the length of the 2-wire line and the way the customer uses the phone, (for example, holding the mouthpiece far away from the mouth);
- 3) the echo due to the mismatch of the hybrid balance circuit impedance and the impedance of the 2-wire line;
- and 4) the automatic gain control (AGC), which is required to compensate for the signal loss in order to achieve a high signal to quantizing noise ratio for better vocoder speech quality. This

automatic gain control amplifies not only the signal, but also the noise and the echo. If the gain is high, it might lead the whole system into singing (or oscillation).

The first source of distortion (i.e. spectral distortion) requires the telephone channel equalization to improve the synthesizer's speech quality. This equalization is nontrivial and unique for the following reasons:

1. Equalization must precede the nonlinear NSP. For example, it is not possible to process the incoming speech with the NSP and then post-process the resulting coefficients with an equalization filter.
2. Equalization can only occur over the frequency range where severe attenuation has not been introduced. If a bandpass filter from 0.3-3.0 kHz has been introduced with 30 dB attenuation in the stopbands, practical considerations such as system noise and integer quantization noise from A/D conversion preclude equalizing the stopband regions.
3. Sophisticated equalization techniques used for digital modems will not work for voice transmission equalization. The high-speed digital modem has an adaptive equalization capability; a series of reference pulses is initially transmitted to the far-end modem, which then adaptively updates the taps or coefficients of an inverse filter or transversal filter until the mean square error between the reference pulses and an

internal standard is minimized. The obvious difficulty with adaptive equalization where voice is involved is that there is no standard for comparison.

The above telephone channel equalization problem was addressed by J.D. Markel and Steven B. Davis at Signal Technology, Inc. [2]. They showed that the channel equalization can be performed by using either a prior long-term spectral characteristic of the speaker, or the population, as a reference signal. Also, it was demonstrated that the equalized vocoded telephone speech is preferred by listeners over nonequalized vocoded telephone speech. Furthermore, listener acceptance of equalized telephone speech improves very nearly to the acceptance level of the reference non-telephone band limited speech by using speaker dependent equalization. Listener preference for speaker independent (population dependent) equalization is slightly lower.

Here we discuss the next three problems--signal loss, echo, and AGC in the loop.

The echo in the telephone line is a very old problem. A great deal of work has been done in developing methods for echo cancellation. These methods range from the simple introduction of a physical loss in the telephone line to the use of adaptive digital filters which adapt to the telephone network. It has been shown that the adaptive digital filters are very effective in satellite telephone networks

where the round trip could be anywhere from 540 ms to 1200 ms [3].

In all the previous studies the vocoder was not present in the loop. Hence, the effect of echo and other telephone distortions combined with the vocoder are not known. Also, it is important to know how the performance of an adaptive digital filter is affected by the presence of the vocoder. A new adaptive filtering algorithm for the echo cancellation in the frequency domain was developed and its performance will be compared to some existing algorithms. The existing algorithms used were the Widrow LMS algorithm and the Gradient Lattice algorithm. The Widrow LMS algorithm was used because of its simplicity and robust performance (which has been shown for many digital signal processing applications including echo cancellation). The Gradient Lattice was chosen because of its orthogonality property which is claimed to result in faster convergence.

The next important point to be considered is the effect of signal loss on the narrowband speech processor. The signal to noise echo requirement for the satisfactory quality of synthetic speech will be defined. Two kinds of noise will be considered here. The first is the telephone channel (2-wire) noise and the second is the quantization noise due to analog to digital conversion. This surely will play an important roll in defining the need or lack of need for the AGC. Even after the need of an AGC is defined, the

AGC issue must be considered very carefully. While the AGC helps to minimize the problem of signal to quantization noise it also introduces other severe problems. A few of these problems are as listed below:

1. With the AGC in the loop the adaptive filter has to keep track of AGC along with the transhybrid response of the hybrid.
2. The present double talker detector algorithm would not work.
3. The AGC not only amplifies the signal, it also amplifies the telephone channel noise and echo.
4. If the AGC gain becomes too large the whole system will go into singing (oscillation).

These problems related to the AGC could be partially solved by feeding the AGC information into the double talker detector (DTK) and adaptive digital filter (ADF) and making AGC slowly varying. "Slowly varying" is a relative term which needs to be defined for our system.

The overall study was performed via a digital simulation of the system shown in Fig 2. The digital simulation was carried out on the STI VAX-11/780 computer.

2. The Test-bed Simulation

2.1 Introduction

Shown in Fig. 1 is a typical analog telephone connection [1]. The near end is connected through a two-wire subscriber loop (or local loop) into the local switch. The local switch attaches to a toll connect trunk which is a two-wire connection into an hybrid. The hybrid then separates the connection into a four-wire transmit and receive path, and the toll switch connects the near and far-end trunks for bi-directional transmission (full duplex). The reason for the four-wire transmission is so that gain can be added to compensate for physical wire loss versus length. At the far end the reverse operations are performed to the subscriber loop. For the Wideband Integrated Network (WIN) there is a combined analog and digital network as illustrated in Fig. 2. The same basic block diagram holds true except that the hybrid now is connected to A/D and D/A converters so that the four-wire transmission line is digital instead of analog. However, this transformation from analog to a digital line has severe effects on the operation of the system.

In Fig. 2 the full duplex speech processor includes an LPC speech analyzer and synthesizer. The telephone signal

conditioner (TSC) block includes those elements in the system needed to compensate for echo introduced by the hybrid and necessary conditioning algorithms to enhance the speech quality of the synthesized speech output. To model the effects of the full-duplex network on the LPC analyzer/synthesizer a full test-bed simulation of the system has been developed. The main purpose of the simulation was to check the effectiveness of different conditioning algorithms on the over-all end-to-end full-duplex operation.

In this section we will first examine the interconnection problems between an analog network, the switch telephone network (STN), and the digital network (WIN). Then we will present an analog/digital simulation of the network which includes those algorithms which are necessary for the system to work in a satisfactory condition.

2.2 Interconnection Problems with STN-WIN

To investigate the problems that arise from such interconnection we follow the block diagram of Fig. 3. The near-end speech input passes first through the telephone mouthpiece which introduces some distortion [2]. Then, the speech input passes through the physical two-wire line, which introduces a loss due to the distance between the

Fig. 3. Block diagram of interfacing the telephone line to a wideband system

calling party (the two-wire portion only) and the location of the hybrid. This loss is a very significant problem as it can range from (a reference of) 0 dB to as much as 20 dB [1]. That means that either we improve the LPC analyzer to perform well for a large dynamic range or we add an Automatic Gain Control (AGC) in the loop to improve the signal-to-quantization noise at the LPC analyzer input.

Another problem is caused by the hybrid which is necessary for converting the standard two-wire line input to the four-wire system (two wires to the analyzer, two wires from the synthesizer). Due to the fact that the impedance balance of the hybrid is a strict function of the loading, and that the loading is a function of the distance from the hybrid to the calling party, this imbalance introduces the most difficult problem in the full-duplex network, the "echo" problem. In general, the hybrid impedance is not perfectly matched and a return signal from the far-end speaker would be fed back through the upper loop to the far-end speaker. This return signal is the "echo".

The inherent impedance mismatches are a function of the calling party's physical two-wire line length to the hybrid. This impedance mismatch causes an echoed signal with relative level of -30 dB at best to -6 dB at worst [3]. To understand the problems caused by the hybrid in this special application, we have to remember that the LPC analyzer needs a high signal-to-quantization noise to achieve acceptable

speech quality. Now, if we assume that the LPC analyzer requires a 0 dB reference for acceptable signal to quantization noise ratio, if both near-end and far-end have a 10 dB loss in the two-wire lines, and if the hybrids each have 10 dB rejection, then there is a state of sustained oscillation (unity loop gain or 0 dB "singing margin" [4]) if we use a standard automatic gain control.

Another problem is caused by the interconnection of the hybrid with the LPC analyzer. For long delays in the four-wire line, the effect of the echo, without the LPC analyzer, is very annoying and is handled generally by adaptive filter algorithms to cancel its effect. With the LPC analyzer in the loop, the effect of the echo becomes even worse because of the inherent nonlinearity in the LPC analyzer. The problem arises particularly during periods when both speakers at the two ends talk at the same time. In these cases the input to the LPC analyzer contains the near-end speaker input and the echo from the far-end speaker, and because of the nonlinearity of the LPC analyzer the result is highly distorted speech at the synthesizer output. We see from the discussion so far that a fast convergence echo cancelling algorithm is needed in our application to minimize the distortion introduced by the LPC analyzer.

Another issue in the full-duplex network is the double-talker detection algorithm. The function of this

algorithm is to detect the presence of near end talker voice input, so that adaptation in the adaptive filter can be halted during such situations to avoid adapting in the wrong direction. Basically the double talker measures the energy at both points at the four-wire side of the hybrid. When the energy ratio between the four-wire transmit and receive side exceeds a certain threshold, the algorithm raises a flag, meaning that the near-end talker is active.

Another issue to consider is the effect of the STN on the input speech signal. Specifically the STN bandpasses the speech, resulting in a range of 300-3200 Hz. Furthermore, the pass band magnitude characteristic may have 10-15 dB of ripple as a function of the input signal energy. This combination of channel distortions and band pass filtering affects the quality of the LPC synthesized speech, severely affecting the pitch and voice/unvoice algorithms. Therefore, it is necessary to improve the existing LPC algorithms to handle such speech signals.

All the issues discussed so far are investigated in this report in terms of algorithms for the telephone signal conditioner (TSC). The algorithms implemented in the test-bed simulation try to solve the problems discussed so far. The effectiveness of those algorithms are checked based on the output speech quality obtained by end-to-end simulations.

In the next subsection the over-all test-bed simulation

is presented.

2.3 The Over-all Test-bed Simulation

In this section we present the over-all test-bed simulation program. The main purpose of this program is to check the interaction between different algorithms needed in the full-duplex communication network with vocoder in the loop. The test-bed program is written in a modular method with a number of "software switches" which can be used to simulate different configurations. This flexibility allows us to check the interaction between the algorithms and isolate and identify the problems caused by such interaction.

Fig. 3 presents a possible block diagram of a full-duplex communication network with vocoder in the loop. The user at site A is connected to site B by a 2-wire line through a local office. The initial connection at site B is a voice connection arrangement (VCA). This device mainly protects the telephone line from any customer equipment. Also, it performs supervisory functions such as a ringing detection, off-hook signaling and automatic seizing of the line. Next the VCA is connected to a hybrid which converts the two-wire line to the 4-wire connection, which in turn is connected to the digital processing system. This system contains a low pass filter for anti-aliasing, automatic gain

control to compensate for telephone line loss and maximize the signal to quantization noise ratio, A/D and D/A, analyzer, synthesizer, double talker detector and echo canceller. Such a block diagram could be simulated entirely by a digital simulator. However, we preferred to divide the system into two parts: 1) The local telephone line from customer mouthpiece to output of VCA; and 2) The wideband system from the 2-wire input/output part of the hybrid at site B to site C. We used the physically available telephone line for part 1, and performed the digital simulation of part 2.

2.3.1 Local Telephone Line Simulation

The input speech data for digital simulation, which is the speaker's speech over the telephone line and the output of the VCA, is collected using the special setup as shown in Fig. 4 and discussed here. Site A and site D called the sites B and C, respectively, through a local office. The output of the VCA at sites B and C were recorded simultaneously on a two-channel tape recorder for future digitization. In order to have natural conversation between the two speakers, the output of each VCA was also connected to the earphone of the first telephone handset. Because of the set-up in Fig. 4, the length of line 1 and line 2 were equal. However, in general this may not be the case. Also,

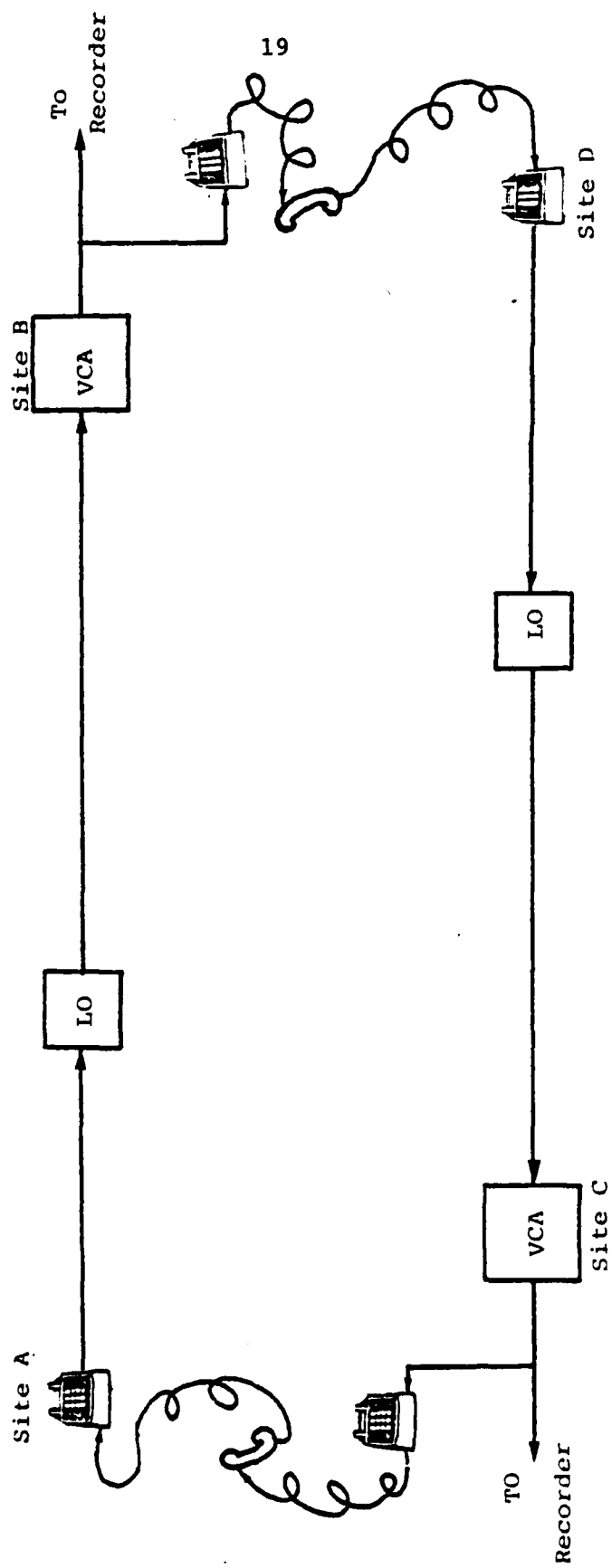


Fig. 4. Data collection over actual telephone line.

the conversation was not fully realistic as it did not include the psychological effects of a satellite delay and an echo problem.

2.3.2 Wideband System Simulation

For the wideband system we simulated digitally the following components:

a) Hybrid

In the hybrid the response from 2-wire to 4-wire and vice versa is not important, and can be approximated by flat unity gain. However, the response from 4-wire receive side to 4-wire transmitter side, which is the function of 2-wire impedance is important. This response is called the transhybrid response. The term transhybrid loss is also used. The transhybrid loss represents the average loss over the frequency. In other words, if the transhybrid loss is 10 dB, then the echo signal at the 4-wire transmitter side is only 10 dB lower than the signal at the 4-wire received side. Previous studies have shown that the average transhybrid loss could be as low as 6 dB in the worst cases [3]. In our case, not only the magnitude of average transhybrid loss, which determines the strength of the echo and hence the perceptual effect, is important, but also the exact response is important in order to evaluate the

performance and requirement of the echo canceller.

In the test-bed simulation, the transhybrid response is simulated by a 64 weight FIR filter. These weights are measured under real telephone lines and from an artificial telephone line where the conditions of loading and line lengths are more controlled. From these measurements a library of 14 different impulse responses has been collected. The user has the option to choose one of them, or by using a software switch change the hybrid response dynamically for each fixed amount of time chosen by the user. In this way a more realistic time variable transhybrid response can be simulated. Details on the measurements of the transhybrid responses and their results are given in Chapter 3.

b) Echo canceller

The echo canceller algorithm is an essential part of the wideband system. Because of the inherent nonlinearity of the LPC analyzer a fast convergent algorithm is necessary.

The Test-bed simulation includes four different algorithms that can be chosen separately by the user: 1) The Widrow LMS algorithm [5], 2) The normalized Widrow LMS algorithm [6], 3) The gradient lattice adaptive filter [7] and 4) The unconstrained frequency domain LMS algorithm (UFLMS) [8].

The Widrow algorithm was inserted in the test-bed simulation for comparison purposes, since it is the most popular algorithm in echo cancelling [9]. The gradient lattice algorithm has been implemented because of its potential fast convergence reported in the literature [7], which happens to be incorrect for our application. The unconstrained frequency domain LMS algorithm was chosen for its efficient implementation and fast convergence compared to the time domain LMS algorithm. The SHARF algorithm was chosen for its simple implementation since the filter model is a recursive filter which can have a low order compared to the FIR model. Since the performance of the SHARF algorithm was very poor compared to the Widrow algorithm we will not elaborate on this algorithm in this report, and the algorithm is not included in the final test-bed simulation.

All the parameters needed to control the different adaptive filter algorithms, such as filter orders and convergence constants, can be chosen by the user during the initialization phase of the simulation program. A complete description of those algorithms is given in Chapter 4.

c) Double talker detector

The main function of the double talker detector is to detect the near-end talking and tell the echo canceller not to adapt when the near-end speaker is talking.

If a double talker is not present, the adaptive

algorithm will drift to the wrong direction during double talking, which gives poor echo reduction, and may distort the signal, depending on the amount of drift present [10].

In the simulation program we have the option to run the experiment with or without the double talker. In the simulation program we have two different double talker algorithms. One is for a time domain algorithm in which the decision is made on a point-by-point basis, the other is for the frequency domain algorithm where the decision is made on a block-by-block basis.

d) Analyzer and synthesizer

The analyzer in the test-bed simulation is an improved version of the standard LPC analyzer. The improvement is mainly in the pitch estimation and voice/unvoiced detection algorithm. Those algorithms are changed to perform better under STN distortions and a large dynamic range of speech input.

All the parameters for the analyzer and synthesizer, such as frame size, pitch detection window, order of the predictor, etc. [for details see 11], can be controlled by the user. Details of the improved algorithm are given in Chapter 5 along with a discussion of the AGC versus improved LPC analyzer.

e) Channel simulation

The channel delay of the digital 4-wire transmission line is simulated by a programmable digital delay. The length of the delay can be controlled by the user. In this study we assume that we are dealing with an ideal digital transmission line and no errors are introduced by the channel.

f) AGC simulation

At first look, an AGC is intuitively very attractive as a solution to the dynamic range problem at the LPC analyzer input. However, after a very careful examination of its interaction with other algorithms, we arrive at the conclusion that the AGC that can be introduced in the full-duplex network, will be very complex and moreover will limit the echo canceller performance to an unacceptable level.

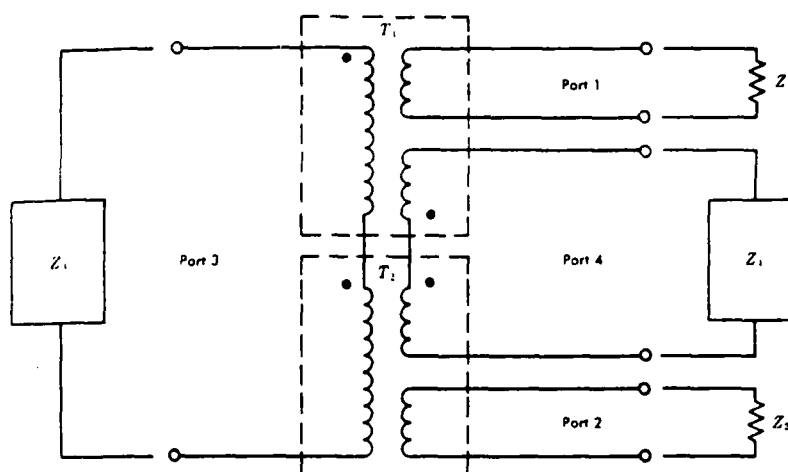
For that reason, an AGC is not included in the test-bed simulation. Instead, we chose the improved analyzer as a solution to the dynamic range problem. A full examination of the AGC is given in Chapter 5.

3. Hybrid Simulation

The hybrid is an analog three port device with one input/output at the 2-wire line, and the receive input and the transmit output at the 4-wire line [Fig. 5]. An exact simulation of the hybrid is very tedious and complex because of its multiple input/output and its inherent nonlinearities. As a first-order approximation, the hybrid responses from the 2-wire line to the 4-wire line and vice-versa can be approximated by a flat unity gain. The standard simulation of the transhybrid response, from the 4-wire line receive to transmit line, is an FIR filter.

In this section we present transhybrid response measurements made on physical telephone lines connected to a physical hybrid. Since in using a physical telephone we did control the loading conditions of the hybrid, the same measurements were repeated using an artificial telephone line, where loading conditions were totally controlled.

The transhybrid measurements were made using standard system identification techniques under linear assumption. To check the accuracy of this assumption, a new nonlinear frequency domain adaptive filter has been used to check the second-order nonlinearity in the hybrid.



The use of the precisely balanced transformer windings to obtain conjugacy between transmission paths results in the so-called *hybrid circuits*. These can be realized with a single transformer structure, but the impedance levels required are usually inconvenient. The more common realization uses two transformers connected as shown by the simplified diagram of Fig. 2-4. Transformers T_1 and T_2 each consist of at least three tightly coupled windings.

If $Z_1 = Z_2$ and $Z_3 = Z_4$, a proper choice of turns ratios will make port 1 conjugate to port 2, and port 3 conjugate to port 4. That is, if Z_1 is a source delivering power to port 1, a negligible part of this power will be received by impedance Z_2 and vice versa. Power flowing into the circuit at either port 1 or port 2 will be delivered to impedances Z_3 and Z_4 equally.

In one practical application, Z_3 is a bilateral two-wire line, and Z_4 is a fixed network whose only function is to match Z_3 and provide the necessary conjugacy. Impedances Z_1 and Z_2 represent a four-wire line using separate pairs for the two directions of transmission. The terms *trans-hybrid loss* and *through-balance* are used to describe the effectiveness of this circuit. Losses of 50 dB between impedances Z_1 and Z_2 are realizable. In central offices where Z_3 is different for every call that is set up, much lower values are common.

Fig. 5. Hybrid circuits using two transformers and its description. (From Transmission Systems for Communication, Bell Telephone Laboratories)

3.1 Transhybrid Measurements under Linear Assumption

To measure the transhybrid response of the hybrid a standard system identification set-up has been built. A user at site A called to site B, and the hybrid at site B was connected to the phone line through a VCA (Voice Connection Arrangement) as shown in Fig. 6. Bandlimited white noise was used as the input to the 4-wire receive side of the hybrid. The user at site A was silent to insure that the signal received at the 4-wire transmit side was only the leakage signal. The input and output signals were digitized simultaneously by a 2-channel A/D. Now the problem reduces to a system identification problem, where the input and the output of a black box is known and the problem is to find the transfer function of the black box.

For system identification we used three different algorithms: the standard Widrow LMS algorithm, a new unconstrained frequency domain algorithm for FIR filter representation, and one which uses the sequential regression algorithm for IIR filter representation. We ran these algorithms on the data collected from the set-up described above. The output of each algorithm was the identified impulse response for the FIR algorithms or the transfer function for the IIR algorithm. Another result from the algorithms was the SNR between the desired response signal and the error obtained by the adaptive filters. Since the SNR achieved by the IIR model was poor compared to the FIR model we continued our measurements on the FIR model only.

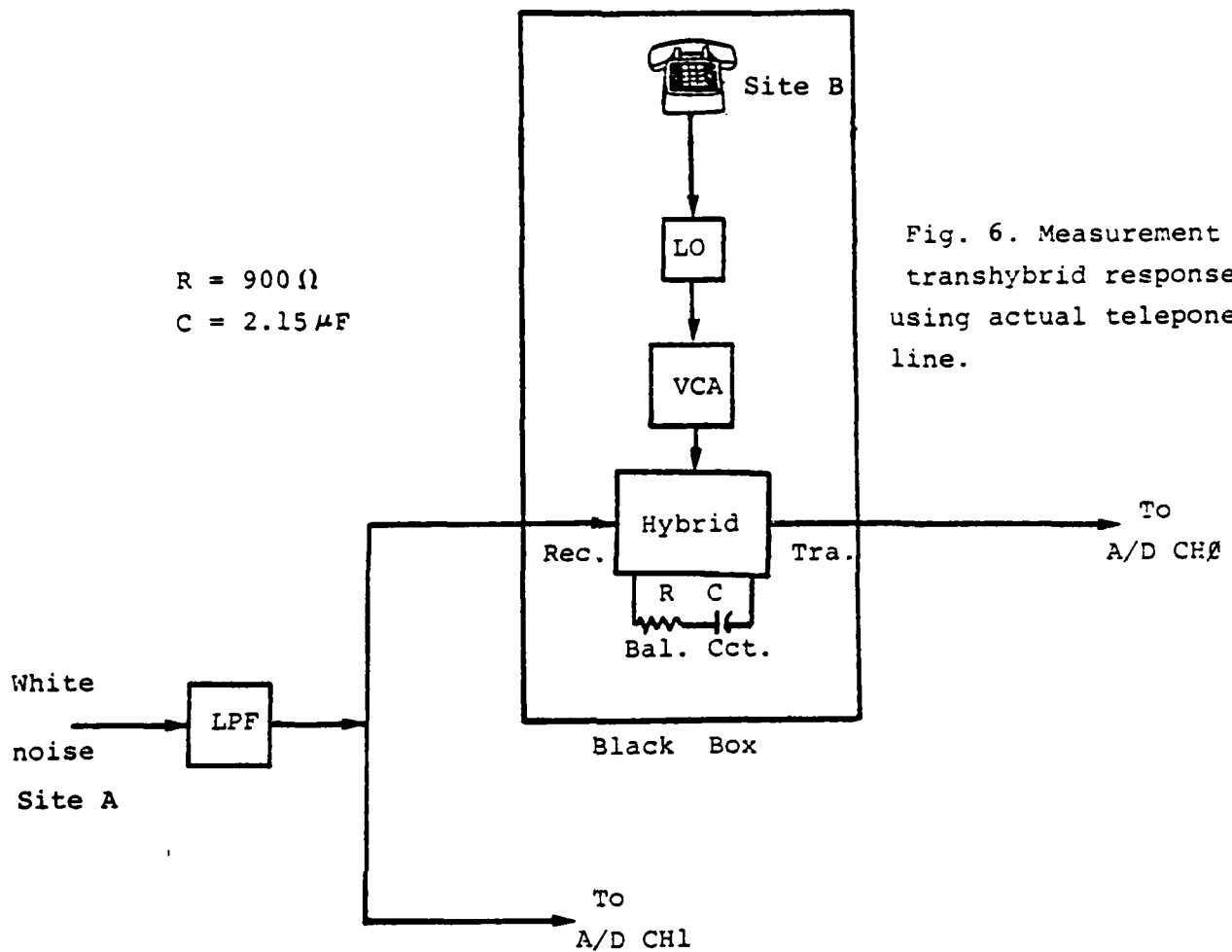


Fig. 6. Measurement of transhybrid response using actual telephone line.

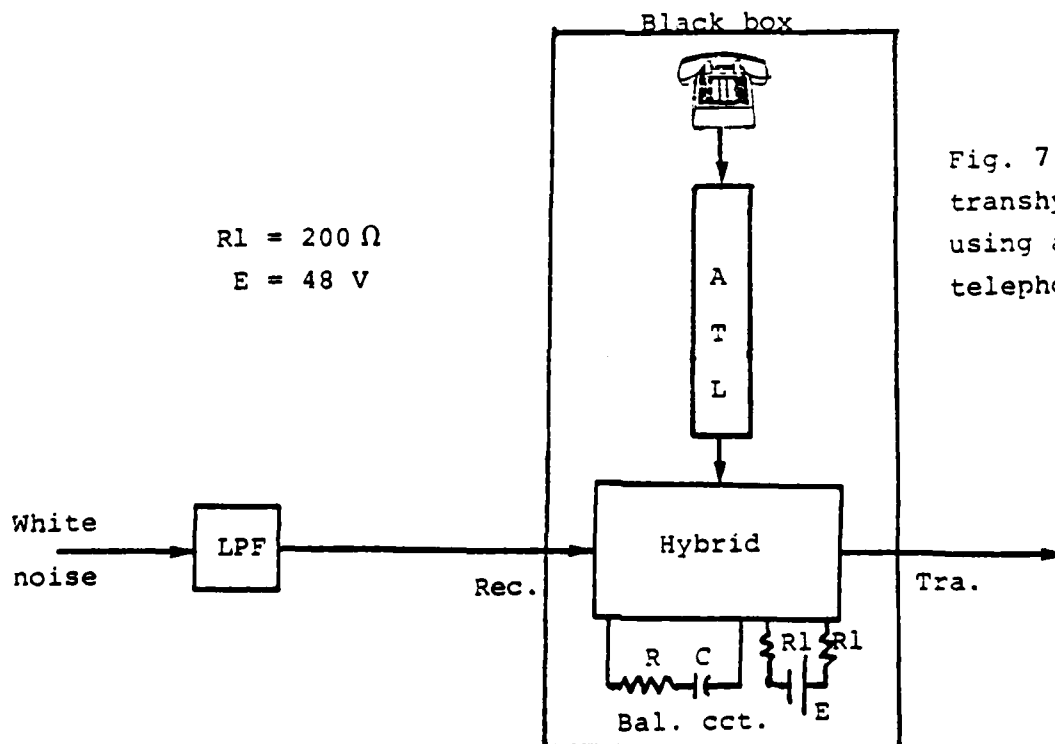


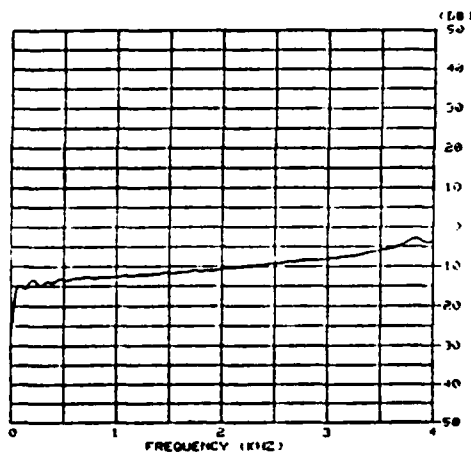
Fig. 7. Measurement of transhybrid response using artificial telephone line.

Details on the algorithms used in those measurements will be given with description of the echo cancelling algorithms.

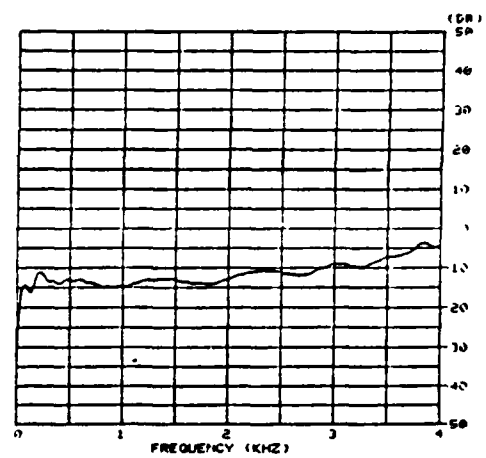
The problem in using the physical telephone for the measurements was that no information was available about the length of the line, i.e., loaded or nonloaded etc. Therefore, a similar experiment was performed under controlled conditions. The black box in Fig. 6 was replaced by the black box in Fig. 7, where an artificial telephone line was used for simulating different length telephone lines. A H88 and D66 loading were used in the loaded lines.

An example of the transhybrid response obtained from the physical telephone lines is given in Fig. 8. Fig. 8a represents the transhybrid response for an unloaded line, and Fig. 8b the transhybrid response for an actual loaded phone line. In Fig. 9 we have a sample of transhybrid responses under different conditions. Fig. 9a and 9b are samples of unloaded artificial line with 5000' and 15,000' respectively. Fig. 9c and 9d present the transhybrid response with different length and different loading.

Since with the artificial telephone line we can control both the length of the lines and the loading conditions, we use it to build a library of different transhybrid responses. The library includes 14 different responses under different conditions. The library is built in a format such that from the test-bed simulation program any one of the fourteen responses can be chosen to simulate the



(a)



(b)

Fig. 8. Transhybrid response with the actual telephone lines.

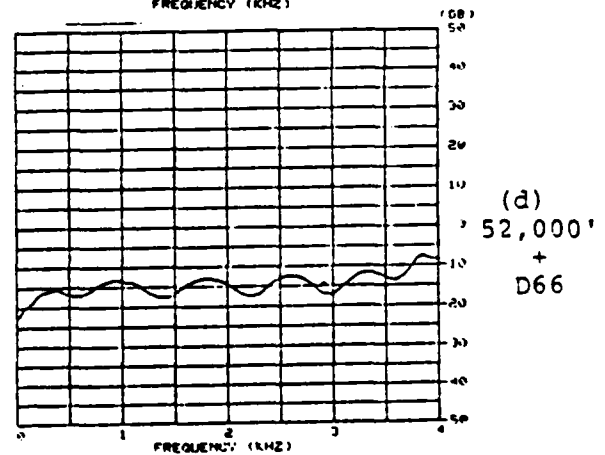
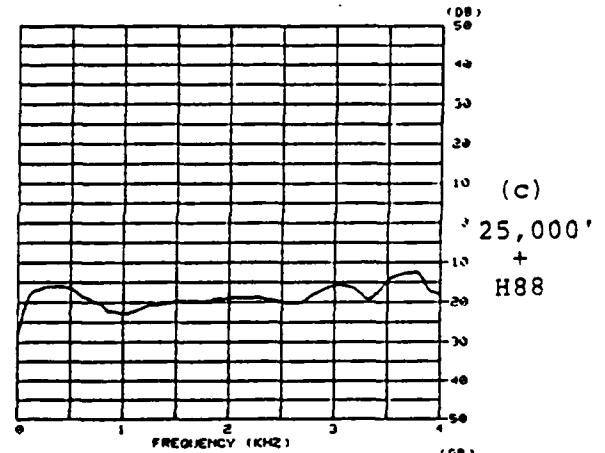
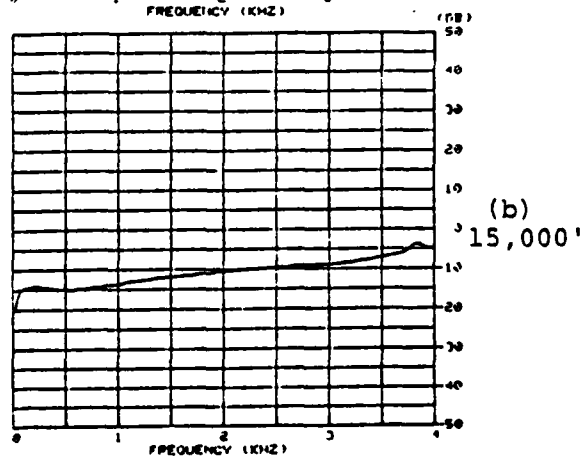
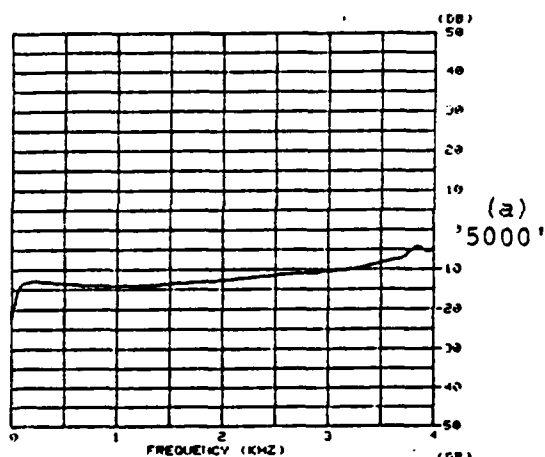


Fig. 9. Transhybrid response with the simulated telephone lines.

transhybrid response. The user can specify different transhybrid responses for the near-end and far-end hybrids. More than that, the user by a software "switch" can dynamically change the transhybrid response during the simulation, and the timing step for those changes can be also controlled by the user. For example the user can specify that every 1 sec the transhybrid response will be changed by another response from the library.

In system identification techniques many measures are used in checking their performances. One such measure is the signal-to-error ratio, which is the ratio between the mean squares output of the system to the mean squares error between the system and its model, when both the system and its model are driven by the same white noise input. For a 64 weight FIR filter the signal-to-error ratio was around 25-28 dB; a longer FIR filter did not achieve better results, and with 32 weights the signal-to-error ratio was lower by one to two dB's. Since we use a 12-bit A/D in our simulation, it means that under linear assumption, the maximum echo reduction that we can achieve is about 25-28 dB.

In the next section we present the measurement of the transhybrid response under nonlinear conditions. Our aim was to check the possibility of simulating a more accurate transhybrid response.

3.2 Transhybrid Response under Non-linear Assumption

One of the major difficulties in dealing with nonlinear system identification is the lack of a unified mathematical theory for representing various nonlinear characteristics. There are, however, a number of representations for nonlinear systems identification purposes. The quality of those representations depends on the kind of the nonlinearity in the system. A well known representation is the Volterra series, with which the least squares technique can be easily applied to nonlinear systems.

The time domain nonlinear systems algorithm that we will describe here is due to Roy and Sherman [12]. Its main drawback is the amount of computations needed. To overcome this complexity a new frequency domain nonlinear algorithm has been developed which reduces the computation by an order of N , where N is the filter length.

Since the time domain nonlinear algorithm is much more easily explained, and to catch the idea of the method, we will present here only the time domain algorithm. The details of the frequency domain algorithm are given in NSC NOTE 147 [13]. Both algorithms converge to the same result. The only advantage to the frequency domain algorithm is the reduction in computation and its fast convergence compared to the time domain algorithm.

3.2.1 Time Domain Nonlinear Adaptive Filter

The nonlinear system identification algorithm described in [12] can be readily interpreted in an adaptive filtering notation. The input-output relationship of a nonlinear system can be expressed explicitly as a Volterra series [14, 15, 16]

$$y(t) = \int_0^{\infty} h_1(\tau)x(t-\tau)d\tau + \int_0^{\infty}\int_0^{\infty} h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1d\tau_2 \\ + \dots + \int_0^{\infty} \dots \int_0^{\infty} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i)d\tau_i + \dots \quad (3.1)$$

The n -th order Volterra kernel $h_n(\tau_1, \tau_2, \dots, \tau_n)$ represents the weighting function of the n -th degree. Thus the n -th order term is an n -fold convolution integral. If we assume that the system is stable and has a finite memory, the system can be approximated by its sampled data form; the output can be written as

$$y(n) = \sum_{i=0}^{N-1} h_1(i)x(n-i) + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} h_2(i,j)x(n-i)x(n-j) + \\ + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} h_3(i,j,k)x(n-i)x(n-j)x(n-k) + \dots \quad (3.2)$$

The objective of the nonlinear adaptive filter is to find the system models $h_1(i)$, $h_2(i,j)$, ..., which minimize the output mean square error. The system can be found by standard least squares technique as done in the linear LMS algorithm. For practical reasons only the quadratic form

will be developed here but the same procedure can be extended to higher orders. For the second order case the output can be written in the following form:

$$y(n) = \underline{y}^T(n) \underline{h} + v(n) \quad (3.3)$$

where $v(h)$ is random additive noise,

$$\underline{y}^T(n) = [x(n), x(n-1), \dots, x(n-N+1), x^2(n), x(n)x(n-1), \dots, x^2(n-N+1)], \quad (3.4)$$

and

$$\underline{h}^T = [h_1(0), h_1(1), \dots, h_1(N-1), h_2(0,0), h_2(0,1), \dots, h_2(N-1, N-1)] \dots \quad (3.5)$$

The filter weights adaptation equations for the k th iteration are

$$h_1^{(k)}(i) = h_1^{(k-1)}(i) + \mu e(n)x(n-i) \quad (3.6)$$

$$h_2^{(k)}(i,j) = h_2^{(k-1)}(i,j) + \mu e(n)x(n-i)x(n-j) \quad (3.7)$$

where μ is the convergence constant. The block diagram of the time domain nonlinear algorithm is presented in Fig. 10.

From equations (3.3) to (3.7) we see that the number of multiply-adds for each output point or iteration is of an order of N^2 or $O(N^2)$. For N output points the number of multiply-adds is thus of $O(N^3)$. Because of this high computational complexity this time domain nonlinear filter is not widely used. A new frequency domain nonlinear algorithm [13] achieves the same performance with an order of N^2 for N output points or $O(N^2)$.

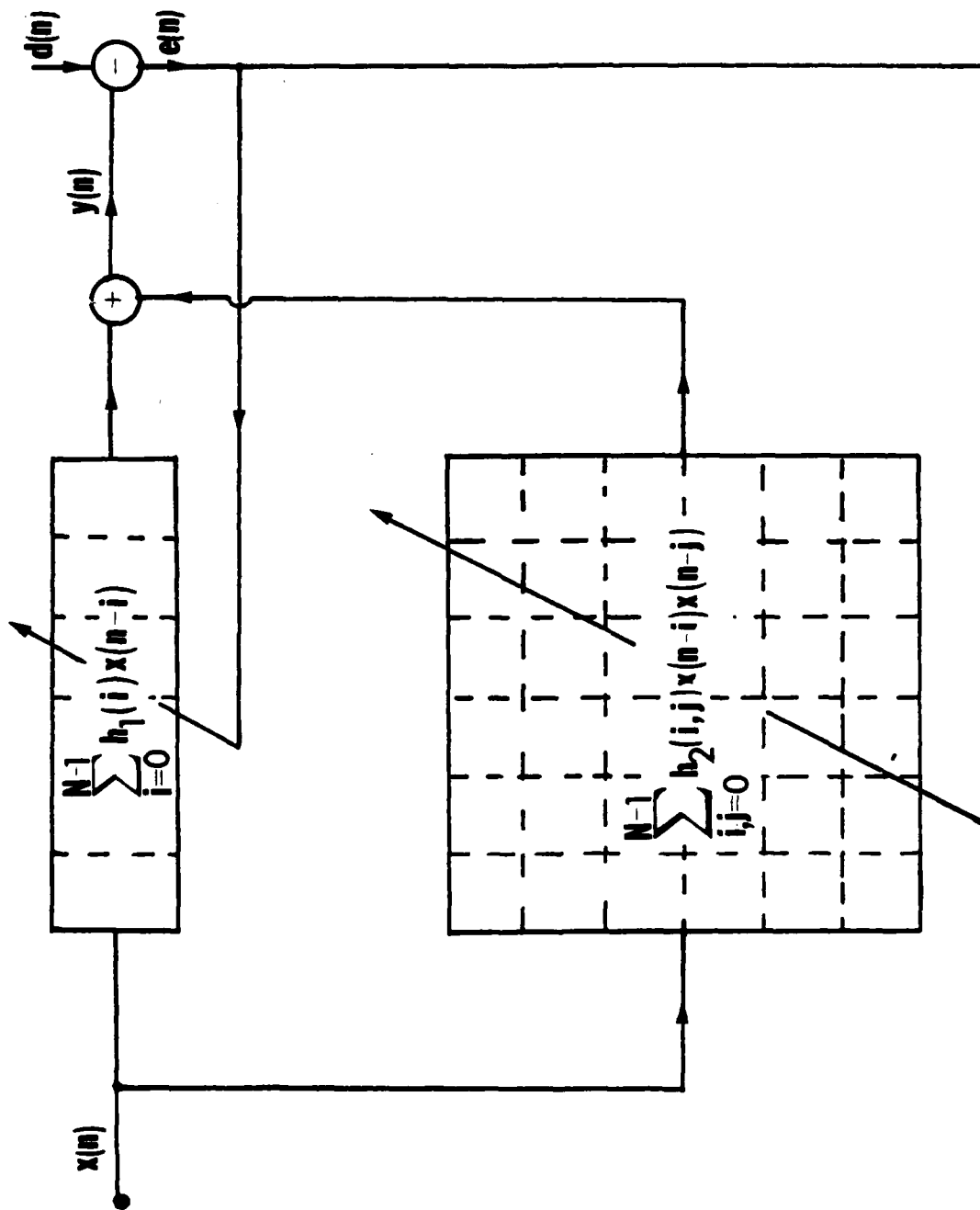


Fig.10. Time domain non-linear adaptive filter

3.2.2. Measurement Results of a Nonlinear Model

For the nonlinear transhybrid response measurement we use the same data collected as described in section 3.1. Instead of using the linear adaptive filter to identify the system, we use a new frequency domain nonlinear adaptive filter as described in [13]. This algorithm was chosen because it has lower computational complexity and faster convergence than the time domain nonlinear adaptive filter.

The results of the nonlinear system identification algorithm are presented in figures 11, 12. Fig. 11 presents the convergence behavior of the linear and nonlinear frequency domain algorithms. The convergence behavior is given in terms of signal-to-error ratio in dB versus number of iterations. From the plots in Fig. 11 we see that the linear algorithm achieves an average signal-to-error of 28 dB. In those results we used a filter of order 70--higher orders did not achieve better results. Fig. 12 presents the actual two-dimensional response as identified by the algorithm.

Those results were obtained from one hybrid, and we don't know the average statistical behavior on different hybrids. Since in the nonlinear algorithm we model only the second order Volterra kernel, it is clear from our results that the main nonlinearity in the hybrid is a second-order nonlinearity.

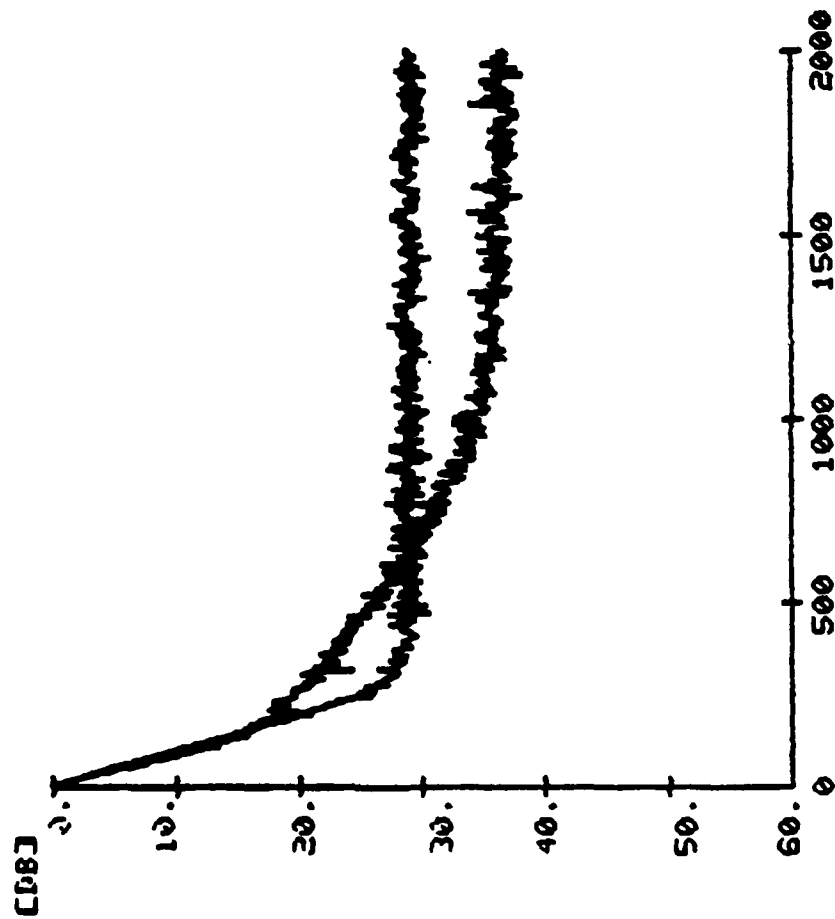


Fig. 11 Convergence behavior of the linear and non-linear frequency domain system identification algorithm for a physical hybrid.

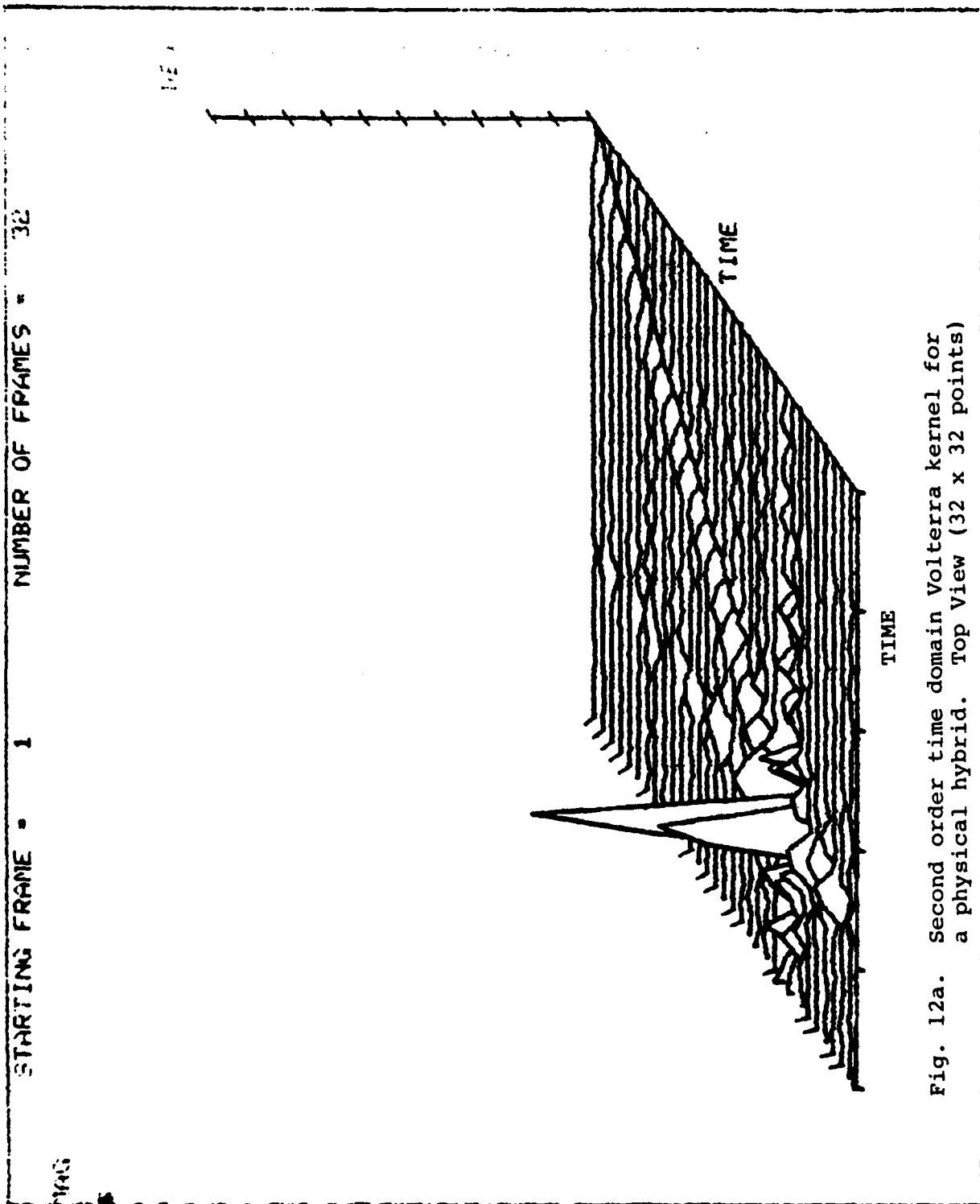
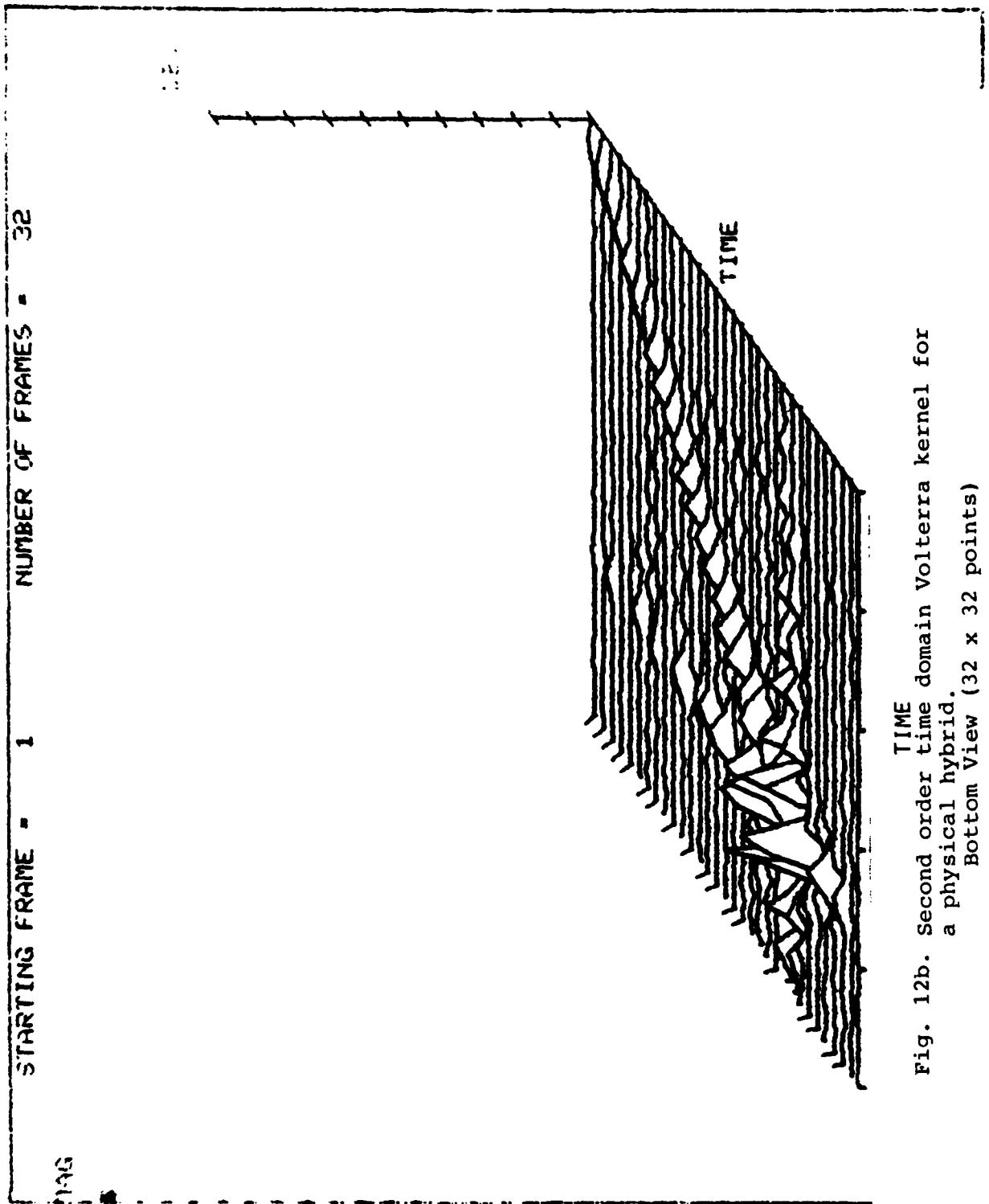


Fig. 12a. Second order time domain Volterra kernel for a physical hybrid. Top View (32 x 32 points)



The main conclusion from this experiment is that the performances of linear echo-canceller are limited by the nonlinearity of the hybrid. The main nonlinearity can be modeled by a second-order Volterra kernel. If for certain applications higher echo reduction than the one achieved by linear algorithm is needed, one can use the nonlinear frequency domain algorithm. The complexity of this algorithm is of $O(N)$, like the Widrow algorithm, where N is the assumed order of the filter. We have to note that with this new algorithm we solved the computation complexity but not the problem of the large amount of memory needed. The amount of memory needed is $O(N^2)$ compared to $O(N)$ in the linear adaptive filter.

4. Echo Cancelling Algorithms

In the study of echo cancelling algorithms for our application, the main goal was to have an algorithm which converges fast enough and still competes with the implementation simplicity of the LMS algorithm. In our application, a faster convergence is needed than in the standard telephone line echo cancelling problem. From experiments presented in Chapter 6, we observe that the effect of the echo is much more destructive with the presence of an LPC analyzer in the loop.

In our search for a fast and yet simple algorithm we checked a number of new recursive adaptive algorithms, proposed recently in the literature. We checked the simple HARF (SHARF) [9] algorithm and the modified hyperstable adaptive recursive filter MHARF [17]. Those algorithms perform very poorly as echo canceller algorithms; for that reason they are not included in the final test-bed simulation program.

The next step was to check nonrecursive adaptive filters with similar complexity as the LMS algorithms. The promising candidates were the gradient lattice algorithm [7] and a new frequency domain algorithm [8]. The gradient lattice algorithm was attractive because of its faster convergence, compared to the LMS algorithm, reported in the

literature. The unconstrained frequency domain algorithm is attractive because of the "pseudo-orthogonality" of the input signal achieved through the use of the D.F.T.

The gradient lattice adaptive filter algorithm is well documented in the literature. Since there are a number of gradient lattice algorithms with minor differences in the literature, we will describe only the gradient lattice algorithm implemented in the test-bed simulation. The new frequency algorithm inserted in the test-bed simulation will be presented in more detail with its convergence rate compared to the LMS algorithm by a simple system identification simulation. After the presentation of the adaptive algorithms inserted in the test-bed simulation we will present the double talker detection algorithm associated with each algorithm.

4.1 The Gradient Lattice Algorithm

The FIR lattice gradient algorithm is derived, as the LMS algorithm, via mean square error minimization criterion. The main difference is that in the gradient lattice algorithm the adaptation is done in two stages. The first stage consists of an adaptive Graham-Schmidt orthogonalization of the input signal. The second stage is a standard LMS algorithm under the assumption that the input signal is already orthogonal. It has been shown that the

gradient lattice achieves faster convergence than the LMS algorithm for highly correlated stationary input signal [7].

The block diagram of the algorithm is given in Fig. 13. The equations controlling the different parameters of the algorithm are given below.

$x(n)$ - input signal to the adaptive filter

$d(n)$ - desired response of the adaptive filter

$b_0(n) = f_i(n) = x(n)$.

The adaptive equations for the i th stage are:

$$K(n+1) = K_i(n) + \frac{\mu_1}{\sigma_i(n)} (f_{i+1}(n)b_i(n-1) + b_{i+1}(n)f_i(n)) \quad (4.1)$$

and

$$g_i(n+1) = g_i(n) + \frac{\mu_2}{\rho_i(n)} (e_i(n)b_i(n)); i=0,1,2\dots N \quad (4.2)$$

where

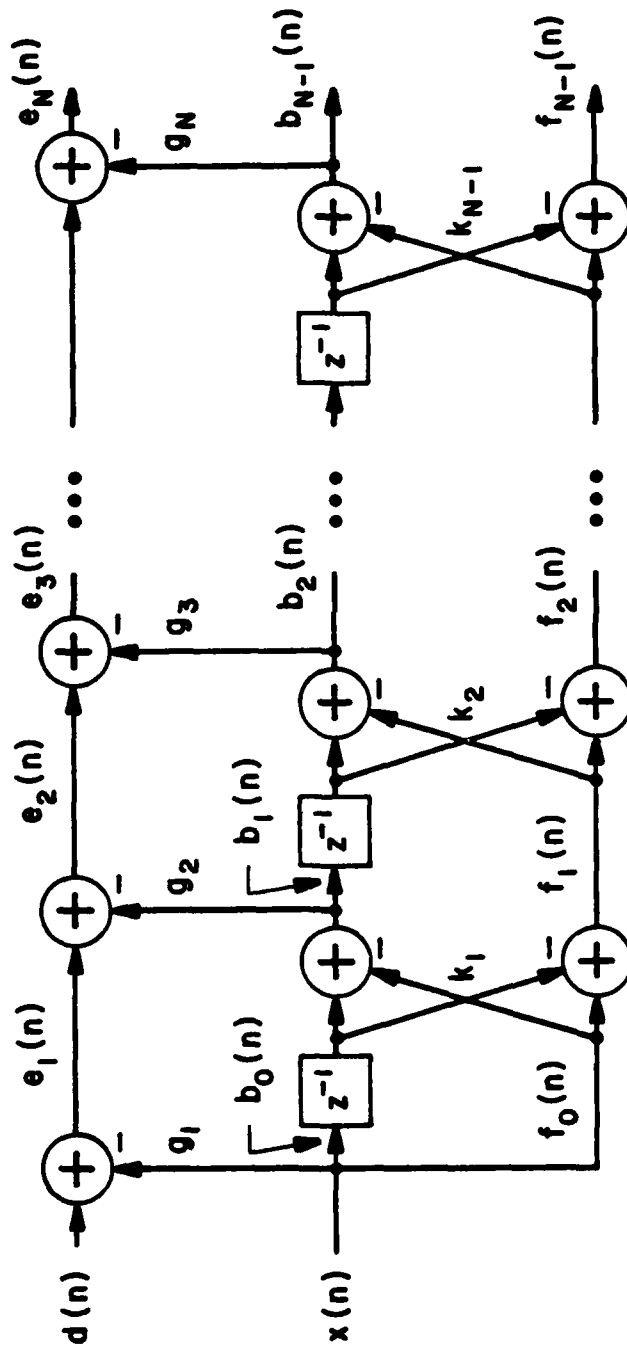
$$\sigma_i(n) = \beta\sigma_i(n-1) + (1-\beta)[b_i^2(n-1) + f_i^2(n)] \quad (4.3)$$

and

$$\rho_i(n) = \beta\rho_i(n-1) + (1-\beta) \cdot b_i^2(n) \quad (4.4)$$

are the input power estimates at the i -th stage. The reflection coefficient $k(n)$ defines the predictor at iteration n , and $g(n)$ defines the filter gain at iteration n . The variables $f(n)$, $b(n)$ and $e(n)$ are the forward prediction errors, backward prediction errors, and the filter error for the i -th stage of the filter at iteration

FIG. 13. GRADIENT LATTICE ALGORITHM



n. μ_1 and μ_2 are the convergence constants, and β is the energy smoothing constant.

This algorithm has been inserted in the test-bed simulation program. The user can specify the order of the adaptive filter N , the convergence constants μ_1 , μ_2 and the energy smoothing constant β .

4.2 The Unconstrained Frequency Domain LMS Adaptive Filter

4.2.1. Introduction

A great effort has been made to develop efficient spectral techniques based on the FFT. Many techniques have been proposed [18,19,20,21] and although they achieve good performance for very restricted applications, they converge to a biased suboptimal solution. For instance, two earlier frequency domain approaches by Dentino et al. [19] achieve good performance as a line-enhancer, and the algorithm by Watzner and Schwartz [21] is specially designed for the isolated training pulse situation for which there is very limited use. Although the use of frequency domain is very attractive (because of the FFT), the key difficulty that has prevented prior work on adaptive filtering from being effective is that the filter must perform linear convolutions whereas the FFT is intrinsically suited for circular convolution.

Recently a more general frequency domain adaptive filter algorithm [FLMS], which converges to the optimal (Wiener) solution, was proposed by Ferrara [22]. The FLMS is efficient for a large number of taps and can be used in general adaptive filtering applications. However, the major drawback of the FLMS algorithm is its slow convergence for highly correlated input signals, as in the case of the time domain LMS algorithm. The FLMS algorithm requires five DFT's in every iteration, two of them are needed to impose time domain constraint in which the last N points of the time domain impulse response are made to be zero.

In this section we introduce a new frequency domain adaptive filtering algorithm that converges to the optimal Wiener solution without the need for any constraints. We demonstrate that the proposed algorithm achieves both faster convergence and reduced complexity compared to the FLMS algorithm. This unconstrained frequency domain algorithm (UFLMS) with normalized convergence constant, achieves faster convergence for an input signal whose covariance matrix has highly disparate eigenvalues.

The UFLMS is presented in section 4.2.2. In 4.2.3. the algorithm with an adaptive convergence constant is introduced and the performance of the UFLMS is compared to the LMS algorithms in a system identification simulation. In 4.2.4. the complexity of the proposed algorithm is described in terms of number of multiply-adds and storage

required. The proof of convergence of the UFLMS to the Wiener solution is given in [8] and will not be given in this report.

4.2.2. The Frequency Domain Adaptive Filter

To simplify the presentation of the frequency domain algorithm, a brief review of the LMS algorithm (Fig. 14) will be given.

The LMS algorithm is a time domain adaptive filter. For each iteration, the weights of the transversal filter \underline{w}_j are adapted according to the equation

$$\underline{w}_{j+1} = \underline{w}_j + 2\mu e_j \underline{x}_j \quad (4.5)$$

where \underline{x}_j is the state vector of input samples stored in the adaptive filter,

$$\underline{x}_j^T = (x_j, x_{j-1}, \dots, x_{j-N+1}) \quad (4.6)$$

The error at the j th iteration is e_j and is defined by

$$e_j = d_j - y_j \quad (4.7)$$

where

$$y_j = \underline{x}_j^T \underline{w}_j \quad (4.8)$$

and d_j is the desired response of the adaptive filter.

In the time domain adaptive filter, e_j and y_j are scalars. In the frequency domain adaptive filter the output of the filter and the error are vectors. New definitions of the input/output are needed in the frequency domain adaptive filter. Capital letters

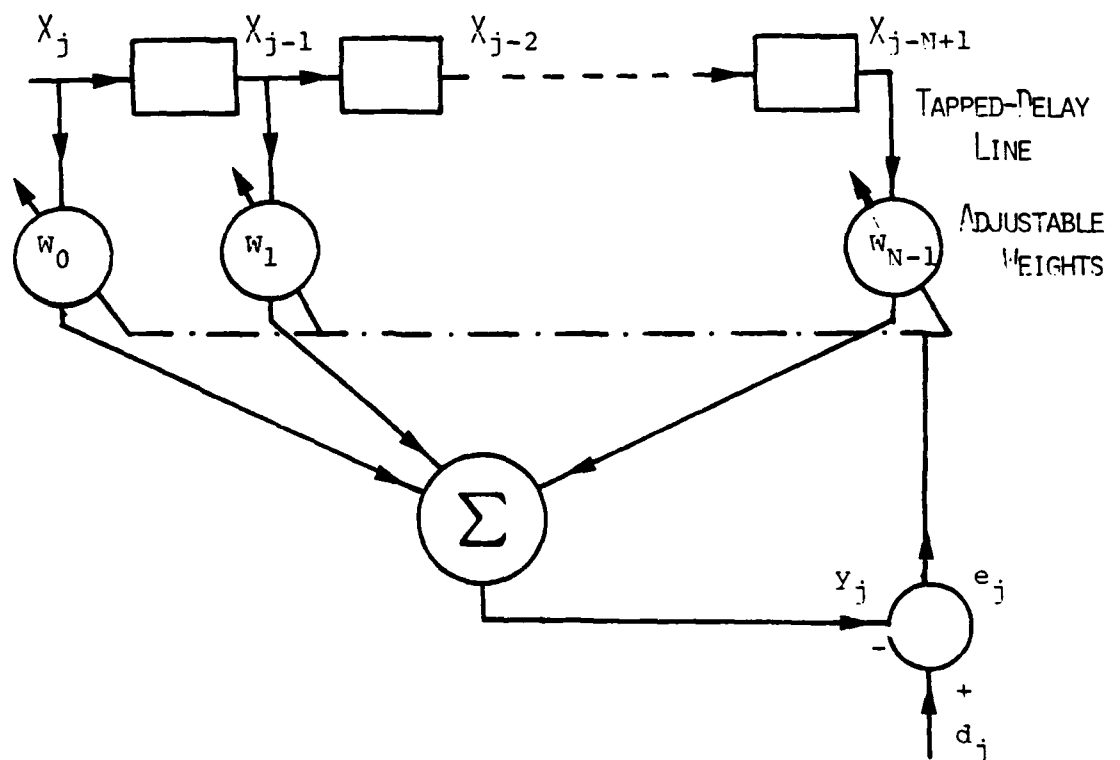


FIG. 14. TIME DOMAIN LMS ALGORITHM

will denote the frequency domain variables, lower case letters, the time domain variables and underlined letters denote vectors or matrices.

The UFLMS algorithm is based on the well known "overlap-save" method used in fast convolution [20]. To clarify the input/output formulations and notations used in the derivation of the UFLMS algorithm, we first introduce the "overlap-save" method using matrix notation.

We assume that the order of the digital filter is less than or equal to $N + 1$. We define a $2N$ impulse response vector h in the following way:

$$\begin{aligned} h(i) &= h(i) & i &= 0, 1, \dots, N \\ h(i) &= 0 & i &= N+1, \dots, 2N-1 \end{aligned} \quad (4.9)$$

The input data stream $x(n)$ is segmented into $2N$ point vectors with N points of overlap in the following way:

$$\begin{aligned} x_K(n) &= x(kN + n) & n &= 0, 1, \dots, 2N-1 \\ & & K &= 0, \infty \end{aligned} \quad (4.10)$$

Using the matrix notation of circular convolution and by dropping K for simplicity in notation, the output vector y_k will be

$$\begin{array}{cccccc}
 y_{2N-1} & x(0) & x(1) & \dots & x(2N-1) & 0 \\
 y_{2N-2} & x(2N-1) & x(0) & x(1) & \dots & x(2N-2) & 0 \\
 y_{2N-3} & x(2N-2) & \dots & \dots & \dots & \dots & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 y_1 & & & & & & h_1 \\
 y_0 & x(1) & x(2) & & x(0) & & h_0
 \end{array} \quad (4.11)$$

We can see from inspection that since the first $N-1$ points of the vector \underline{h} are zero, and the last N point in output vector \underline{y} are the result of linear convolution. In the "overlap-save" method this matrix operation is done using FFT techniques which give a high computational efficiency. We see that in each iteration k we have $2N$ data inputs and N data outputs, for that reason the input data is overlapped by N points.

Now we will introduce different matrix operations that we will use in the derivation of the UFLMS algorithm.

Notation and Definitions

Let \underline{F} be a symmetric $2N \times 2N$ matrix whose elements are $F_{kj} = \exp(-i(2\pi/2N)kj)$, $k, j = 0, 1, \dots, 2N-1$, where i is the square root of -1 . When \underline{F} operates on a column vector of order $2N$ the result is a column vector representing the DFT of the original vector.

Let \underline{F}^{-1} be the inverse of matrix \underline{F} , one can show that \underline{F}^{-1} is a symmetric $2N \times 2N$ matrix whose elements are $F_{kj}^{-1} = 1/2N \exp(i(2\pi/2N)kj)$. It can be shown that \underline{F} and \underline{F}^{-1} have the following properties:

$\underline{F}^\dagger = 2N \underline{F}^{-1}$ and $(\underline{F}^{-1})^\dagger = 1/2N \underline{F}$, where \dagger denotes the transpose complex conjugate.

Let \underline{x}_k be the circulant matrix defined by equation (4.10) and (4.11) for the iteration k .

Now define:

$$\underline{x}_k = \underline{F} \underline{x}_k \underline{F}^{-1},$$

since \underline{x}_k is a circulant matrix, \underline{x}_k is a diagonal matrix whose elements are the DFT output of the first row of the circulant matrix \underline{x}_k (for details see [25]). Using the properties of the matrices \underline{F} and \underline{F}^{-1} we have $\underline{x}_k^\dagger = \underline{F} \underline{x}_k^\dagger \underline{F}^{-1}$.

Let \underline{h} be a $2N \times 2N$ windowing matrix whose lower $N \times N$ right corner is the identity matrix:

$$\underline{h} = \begin{pmatrix} \underline{0} & \underline{0} \\ \underline{0} & \underline{I} \end{pmatrix}.$$

By inspection, $\underline{h} \underline{h} = \underline{h}$. We define

$$\underline{H} \triangleq \underline{F} \underline{h} \underline{F}^{-1}.$$

Since \underline{h} is diagonal, \underline{H} is a circulant matrix whose first row is the inverse DFT of the vector $(0 \dots 0 \ 1 \ 1 \dots 1)$

The following equalities hold for \underline{H} :

$$\underline{H} \underline{H} = \underline{E} \underline{h} \underline{E}^{-1} \underline{E} \underline{h} \underline{E}^{-1} = \underline{E} \underline{h} \underline{E}^{-1} = \underline{H} \quad , \quad (4.12)$$

$$\underline{H}^\dagger = (\underline{E}^{-1})^\dagger \underline{h}^\dagger \underline{E}^\dagger = \underline{E} \underline{h} \underline{E}^{-1} = \underline{H} \quad . \quad (4.13)$$

Let $\underline{\mu}$ be a $2N \times 2N$ real diagonal positive definite matrix, where the elements on its diagonal represent the convergence constant for each frequency in the UFLMS algorithms to be introduced later.

Let \underline{d}_k , \underline{w}_k , \underline{e}_k , \underline{y}_k be $2N$ point vectors representing the desired response, the impulse response, the error vector and the output vector in the time domain.

Finally, let \underline{D}_k , \underline{W}_k , \underline{E}_k , \underline{Y}_k be the $2N$ points DFT's of the vectors \underline{d}_k , \underline{w}_k , \underline{e}_k , \underline{y}_k respectively.

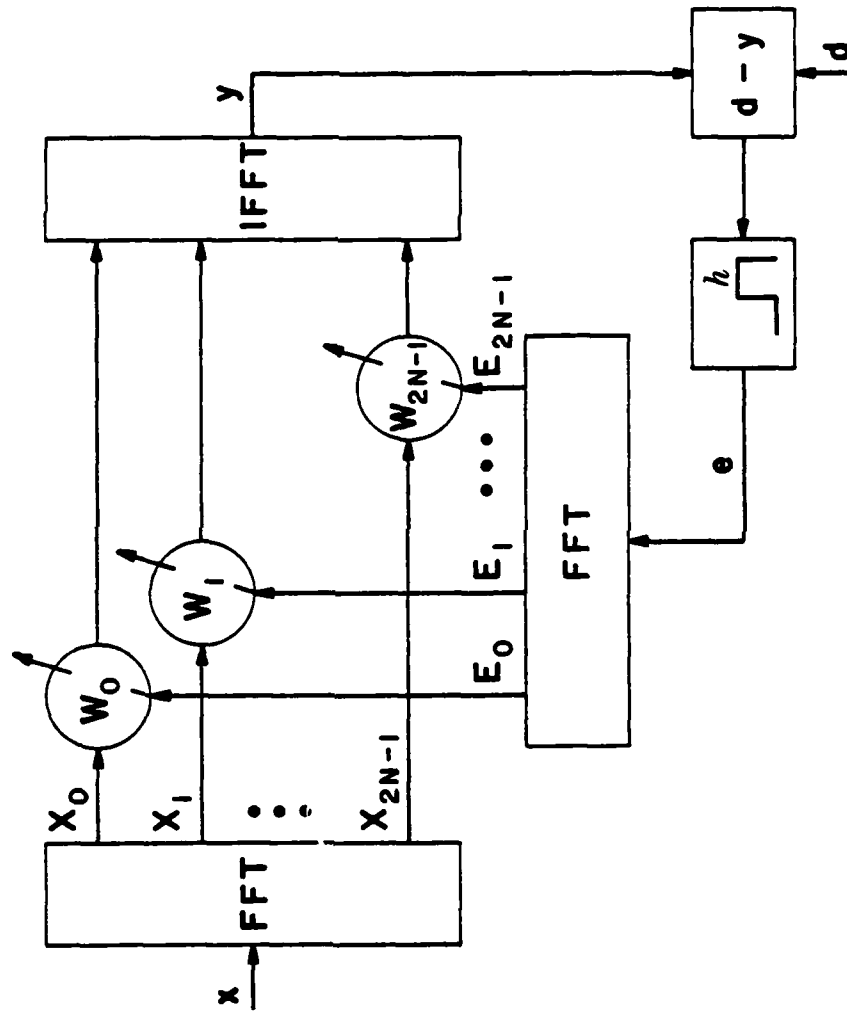
Using the above definitions and the block diagram of Fig. 15, we can present the unconstrained frequency domain adaptive filter. The configuration is based on the "overlap-save" technique [24]. The diagonal matrix \underline{X}_k , derived from the DFT of the $2N$ point input vector at iteration k , is multiplied by \underline{W}_k , the DFT of the impulse response, to give the output \underline{Y}_k ,

$$\underline{Y}_k = \underline{X}_k \underline{W}_k \quad (4.14)$$

The DFT output in the time domain is

$$\underline{y}_k = \underline{E}^{-1} \underline{X}_k \underline{W}_k \quad (4.15)$$

FIG. 15 UNCONSTRAINED FREQUENCY DOMAIN
ADAPTIVE FILTER



The error in the time domain before windowing is

$$\underline{e}_k' = \underline{d}_k - \underline{y}_k \quad (4.16)$$

According to the "overlap-save" method, when the filter length is $N+1$ only the last N points of \underline{y}_k are the result of a linear convolution; then the vector error output must be windowed by \underline{h} to give

$$\underline{e}_k = \underline{h} (\underline{d}_k - \underline{y}_k) = \underline{h} (\underline{d}_k - \underline{F}^{-1} \underline{X}_k \underline{W}_k) \quad (4.17)$$

Since we want to adapt the coefficients in the frequency domain, we take the DFT of equation (4.17) by multiplying both sides by \underline{F} ,

$$\begin{aligned} \underline{E}_k &= \underline{F} \underline{e}_k = \underline{F} \underline{h} (\underline{d}_k - \underline{F}^{-1} \underline{X}_k \underline{W}_k) \\ \underline{E}_k &= \underline{F} \underline{h} \underline{F}^{-1} (\underline{F} \underline{d}_k - \underline{F} \underline{F}^{-1} \underline{X}_k \underline{W}_k) \\ \underline{E}_k &= \underline{H} (\underline{D}_k - \underline{X}_k \underline{W}_k) \end{aligned} \quad (4.18)$$

To derive the adaptive algorithm, we can follow the same steps used to derive the LMS algorithm. Assuming that the input signal is stationary and the identified system is time-invariant, we can show that the expected value of the squared error $\varepsilon \{ \underline{E}_k^2 \}$ is quadratic in \underline{W}_k . By using the gradient method, the algorithm will converge to some \underline{W}^* that achieves the minimum mean square error. The squared error at each iteration is

$$\underline{E}_k^2 = \underline{E}_k^\dagger \underline{E}_k = (\underline{D}^\dagger - \underline{W}_k^\dagger \underline{X}_k^\dagger) \underline{H}^\dagger \underline{H} (\underline{D} - \underline{X}_k \underline{W}_k) , \quad (4.19)$$

as \underline{H} is symmetric and idempotent

$$\underline{E}_k^2 = (\underline{D}_k^\dagger - \underline{W}_k^\dagger \underline{X}_k^\dagger) \underline{H} (\underline{D}_k - \underline{X}_k \underline{W}_k) . \quad (4.20)$$

Assume for now that the weight vector \underline{W}_k is independent of \underline{X}_k . This assumption serves only to motivate the design of the algorithm and is not used in the convergence proof which follows later. The expected squared error will be

$$\begin{aligned} \varepsilon\{ \underline{E}_k^2 \} &= \varepsilon\{ \underline{D}_k^\dagger \underline{H} \underline{D}_k \} - 2\varepsilon\{ \underline{D}_k^\dagger \underline{H} \underline{X}_k \} \underline{W}_k \\ &\quad + \underline{W}_k^\dagger \varepsilon\{ \underline{X}_k^\dagger \underline{H} \underline{X}_k \} \underline{W}_k , \end{aligned} \quad (4.21)$$

where ε denotes expected value.

Equation (4.21) is quadratic in \underline{W}_k . The optimal vector \underline{W}_k^* is obtained by setting the gradient with respect to \underline{W}_k equal to zero,

$$\nabla_{\underline{W}_k}^\dagger \triangleq -2\varepsilon\{ \underline{D}_k^\dagger \underline{H} \underline{X}_k \} + 2 \underline{W}_k^\dagger \varepsilon\{ \underline{X}_k^\dagger \underline{H} \underline{X}_k \} = 0 . \quad (4.22)$$

The UFLMS algorithm uses the gradient method to solve equation (4.22). According to this method the "next" weight vector \underline{W}_{k+1} is equal to the present weight vector \underline{W}_k plus an increment proportional to the negative gradient,

$$\underline{W}_{k+1} = \underline{W}_k - \mu \nabla_{\underline{W}_k} . \quad (4.23)$$

In the UFLMS algorithm the gradient vector is estimated by its instantaneous value at iteration k . As in the LMS algorithm this instantaneous gradient is used to approximate the real gradient. From equation (4.22) the equation controlling the filter weights will be

$$\underline{W}_{k+1} = \underline{W}_k + 2 \underline{\mu} \underline{X}_k^\dagger (\underline{H} \underline{D}_k - \underline{H} \underline{X}_k \underline{W}_k) \quad , \quad (4.24)$$

or by using (4.18), equation (4.24) becomes

$$\underline{W}_{k+1} = \underline{W}_k + 2 \underline{\mu} \underline{X}_k^\dagger \underline{E}_k \quad , \quad (4.25)$$

where the diagonal matrix $\underline{\mu}$ is the convergence constant as in the LMS algorithm. As we shall see, we can choose different convergence constants for different frequencies.

In [8] we prove that the adaptive filter is stable if $\mu(i,i)$ is bounded by the maximum of $(X_{k(i,i)})^2$ for all k . Here we present the algorithm in a simplistic way; in [8] we proved that the algorithms converge to the optimal solution under the following conditions.

1. The input signal to the adaptive signal is ergodic, stationary and bounded.
2. The covariance matrix of the input signal has at least rank $N+1$.
3. The identified system is time invariant.
4. The order of the identified system is less than or equal to $N+1$.

4.2.3. Simulation results

From [8], we have to choose $0 < \mu_{ij} < 1/M_i$, where M_i is the upper bound for the energy at frequency i . Since in practical applications, we do not know those bounds, we choose to estimate M_i by normalizing a constant convergence factor α by an estimate of the energy at the i -th frequency. This normalization is similar to the process used in the lattice adaptive filter [7]:

$$\mu_{k+1}(i) = \frac{\alpha}{z(i)_k} \quad , \quad (4.26)$$

where

$$z(i)_k = (1 - \beta) z(i)_{k-1} + \beta X_k(i) X_k^*(i) \quad (4.27)$$

where i is the frequency index, α is the normalized convergence factor for all the frequencies and β is the energy smoothing constant for all frequencies. As in the lattice adaptive filter algorithm, different smoothing and normalization algorithms can be applied depending on the applications.

To illustrate the rapid convergence rate of the UFLMS algorithm for highly correlated input signal, two simulations of system identification are discussed, one with uncorrelated input signal and the other with a highly correlated signal.

A 32 weight F.I.R. filter was chosen as the system to be identified, with the actual values of the impulse response given in

	NUMERATOR	DENOMINATOR
1	0.230847E+00	0.100000E+01
2	-0.102171E+00	0.000000E+00
3	0.383698E-01	0.000000E+00
4	-0.295674E-01	0.000000E+00
5	-0.575074E-02	0.000000E+00
6	-0.693523E-02	0.000000E+00
7	-0.295320E-01	0.000000E+00
8	-0.690188E-02	0.000000E+00
9	-0.503678E-01	0.000000E+00
10	0.102334E-01	0.000000E+00
11	-0.448287E-01	0.000000E+00
12	-0.354443E-01	0.000000E+00
13	-0.331595E-01	0.000000E+00
14	0.515750E-02	0.000000E+00
15	-0.419697E-01	0.000000E+00
16	0.518027E-02	0.000000E+00
17	-0.163817E-01	0.000000E+00
18	-0.926803E-02	0.000000E+00
19	-0.896433E-03	0.000000E+00
20	-0.948155E-02	0.000000E+00
21	-0.180632E-02	0.000000E+00
22	-0.774926E-03	0.000000E+00
23	-0.647487E-02	0.000000E+00
24	0.329426E-02	0.000000E+00
25	-0.438749E-02	0.000000E+00
26	0.927637E-03	0.000000E+00
27	-0.200423E-04	0.000000E+00
28	-0.152551E-02	0.000000E+00
29	0.340074E-02	0.000000E+00
30	-0.310239E-02	0.000000E+00
31	0.551548E-02	0.000000E+00
32	-0.336223E-02	0.000000E+00

TABLE 1. COEFFICIENTS OF 31-TH ORDER ALL
ZERO FILTER (FIR)

Table 1. This F.I.R. filter approximate a transhybrid response [26]. In the first experiment, the input signal $x(n)$ was white random noise uniformly distributed between -1000 and 1000. The desired response d_k was the result of the convolution of $x(n)$ with the F.I.R. filter. Since d_k in our simulation was transformed back from floating to integer representation, the additive noise in this simulation was the quantization noise. Figure 16 presents the block diagram of the computer simulation.

In Fig. 17, we have the convergence of UFLMS and LMS algorithms, presented as the S/N in dB between the energy in the desired response signal to the error signal. We see that both algorithms have almost the same convergence rate as expected from the discussion in [8]. For the LMS algorithm a μ of 0.1×10^{-5} was chosen to achieve the fastest convergence rate. In the UFLMS algorithm $\alpha = 0.4$ and $\beta = 0.8$ were chosen to achieve the same misadjustment error as in the LMS algorithm.

In the second experiment, the input signal was changed to a highly correlated signal by passing the same white noise from the first experiment through a 12th order all pole filter; the coefficients of the filter are given in Table 2. The motivation was to produce a highly correlated signal and with this filter a $\lambda_{\max}/\lambda_{\min}$ of 20 was achieved.

In Fig. 18 the convergence rate with the correlated signal is given for the LMS and UFLMS algorithms. We notice that the convergence of the UFLMS is almost the same as with correlated signal as with the uncorrelated signal. The LMS algorithm has slow

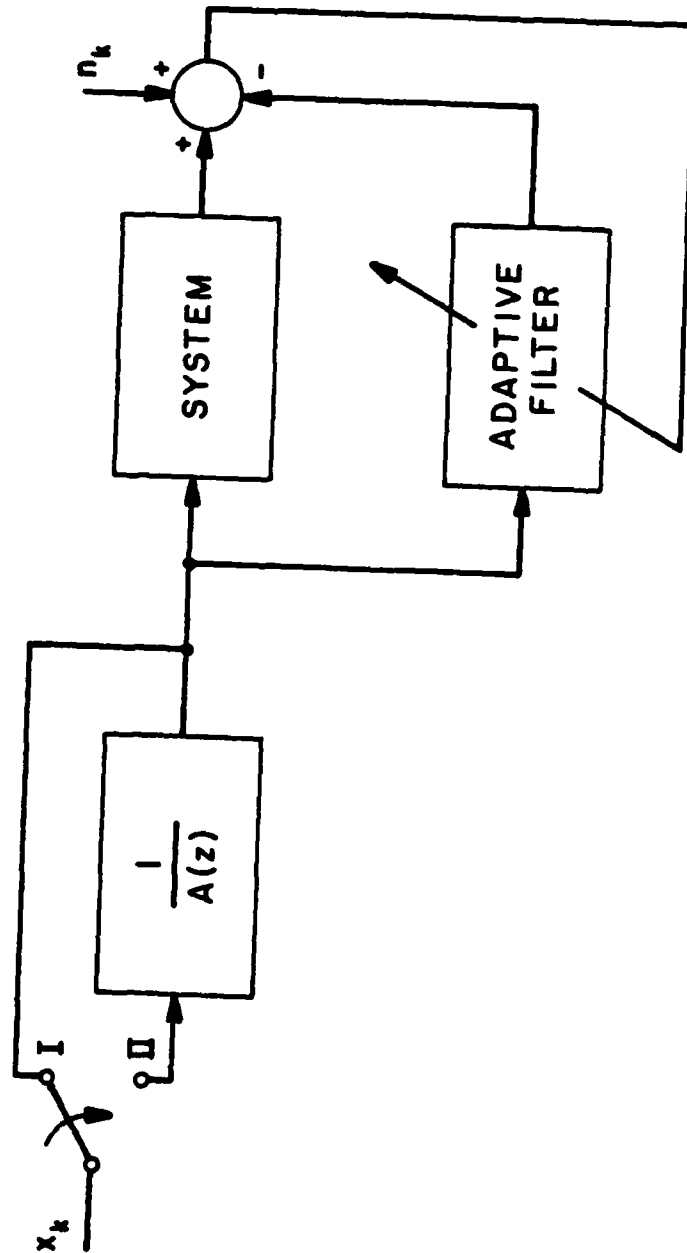


FIG. 16. SIMULATION BLOCK DIAGRAM

	NUMERATOR	DENOMINATOR
1	0.100000E+01	0.100000E+01
2	0.000000E+00	-0.146034E+01
3	0.000000E+00	0.126638E+01
4	0.000000E+00	-0.850541E+00
5	0.000000E+00	0.629542E+00
6	0.000000E+00	-0.497503E+00
7	0.000000E+00	0.273701E+00
8	0.000000E+00	-0.168227E+00
9	0.000000E+00	0.257914E+00
10	0.000000E+00	-0.238396E+00
11	0.000000E+00	0.508109E+00
12	0.000000E+00	-0.379440E+00
13	0.000000E+00	0.204267E+00

TABLE 2. COEFFICIENTS OF 12-TH ORDER ALL
POLES FILTER

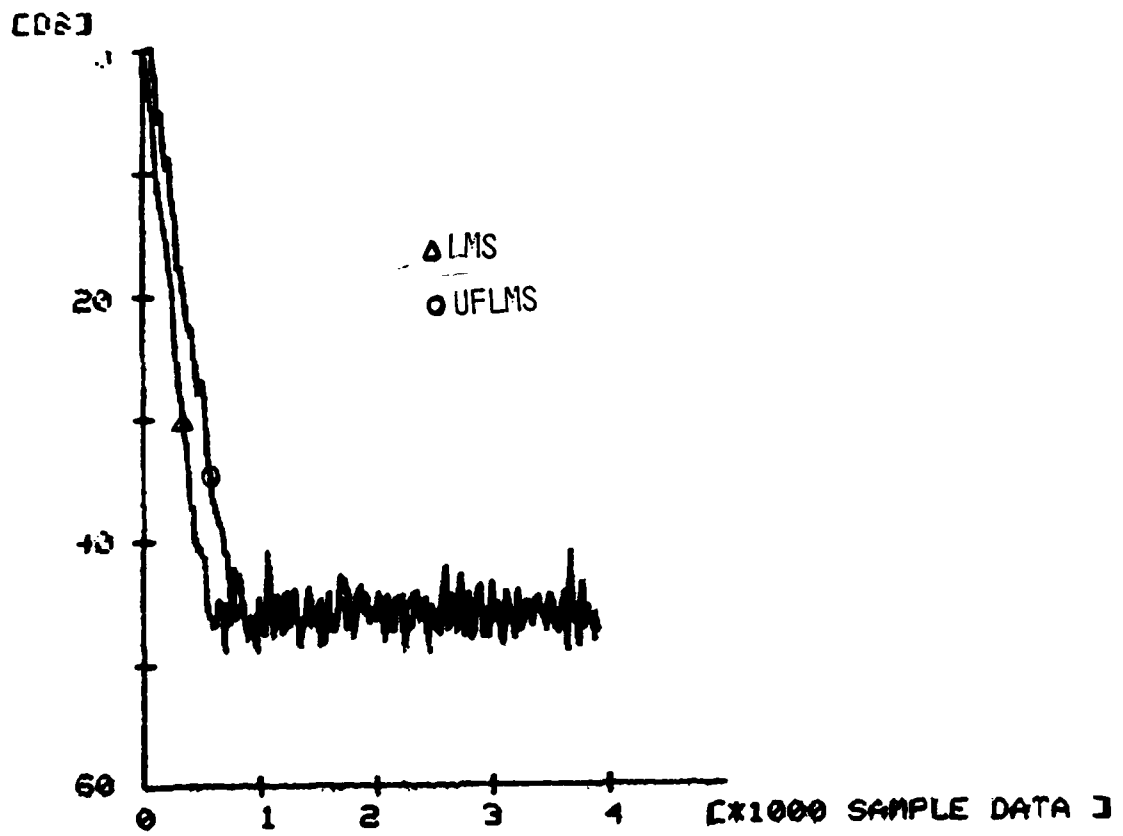


FIG. 17. CONVERGENCE OF LMS AND UFLMS ALGORITHMS FOR UNCORRELATED INPUT.

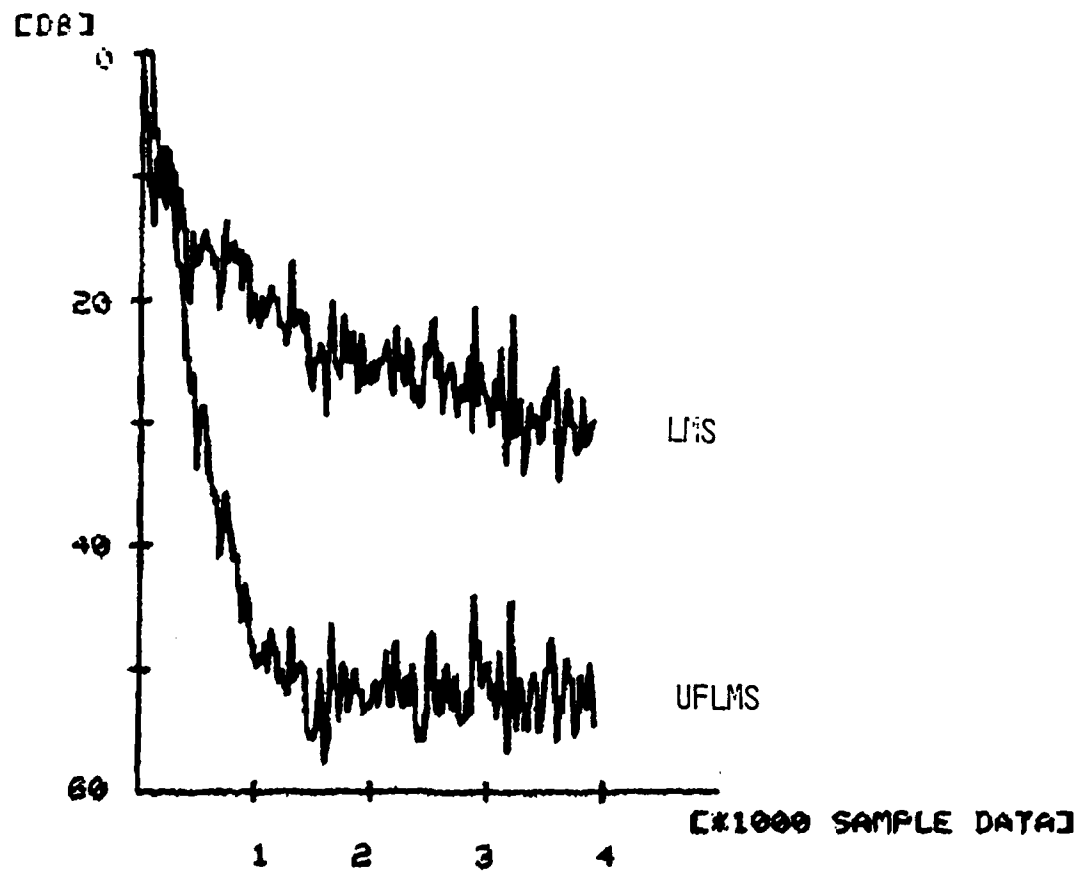


FIG. 18. CONVERGENCE OF LMS AND UFLMS ALGORITHMS FOR CORRELATED INPUT.

convergence with the highly correlated signal. For the LMS algorithm a $\mu = 0.2 \times 10^{-7}$ was chosen to achieve the fastest convergence possible, for the UFLMS algorithm a $\alpha = 0.09$ and $\beta = 0.8$ were chosen to achieve the same misadjustment error. These simulation results illustrate the advantage of the UFLMS algorithm for highly correlated signals over the LMS algorithm.

4.2.4. Complexity of the Algorithm

For large N (filter length -1) the algorithm is very efficient. To filter N points the conventional LMS algorithm requires N iterations or $2N(N+1)$ real multiplies. For the same N points, the UFLMS algorithm requires three FFT's of $2N$ real data points, two $2N$ complex multiplies and two $2N$ real multiplies. For real input data, the output of the FFT is conjugate symmetric so that only the first $N+1$ values need be updated. Furthermore, for real data, the $2N$ point FFT can be realized with an N point FFT and N complex multiplies using an array of N complex points. Each N point FFT requires $N/2 \log N/2$ complex multiplies. Therefore, the number of complex multiplies per block will be $3(N/2 \log N/2 + N)$ for the three FFT's, $2N$ complex multiplies for the complex weighting and adaptation, and $2N$ real multiplies for the convergence constants. For the adaptive convergence constant $5N$ additional real multiplies are required.

Assuming one complex multiply is equivalent to 4 real multiplies, the ratio γ between the UFLMS real multiplies and the LMS real multiplies will be

$$\gamma = \frac{3 \log \frac{N}{2} + 11}{N} \quad (4.28)$$

for constant convergence factor and

$$\gamma^2 = \frac{3 \log \frac{N}{2} + 13.5}{N} \quad (4.29)$$

for the adaptive convergence factor. This ratio is computed for several values of N in the following table

N	γ_1	γ_2
16	1.25	1.4
32	0.72	0.8
64	0.4	0.44
128	0.22	0.24
256	0.12	0.13
512	0.068	0.072
1024	0.037	0.039

From the table we see that for $N = 32$ the proposed algorithm is already more efficient than the LMS algorithm.

The UFLMS algorithm requires 3 real arrays of $2N$ points each for the three FFT's and one $2N$ real data array for the filter coefficients. For the adaptive convergence factor one more array of N real data points is needed. The sine table for the FFT needs an array of $N/4$ real data points. Overall we need about $8N$ points of memory compared to only $2N$ in the LMS algorithm.

We see that what we gain in computation we lose in memory. For special purpose hardware the proposed algorithm is even more efficient since we are working on block operations which can be done very efficiently with array processors.

4.3. The Double Talker Algorithm

The function of the double talker is to detect the presence of near end talker voice input, so that adaptation in the adaptive filter can be halted when a near end talker is detected to avoid adapting the coefficients in the wrong direction. Basically, the double talker detector measures the energy at both points at the four-wire side of the hybrid, and when the energy ratio between the four-wire transmit and receive side exceeds a certain threshold, the algorithm decides that near end talker input is present.

For the LMS and the gradient lattice algorithm a simple double talker detection algorithm taken from [6] is used. It compares the output signal $d(n)$ of the hybrid with the input signal $x(n)$ of the hybrid over the L preceding sample points. If the signal $d(n)$ is greater than one-third the largest absolute (in magnitude) value of $x(n)$ over L preceding sample points, double talking is "detected". The number L is the number of the impulse response of the transhybrid function, chosen by the user. In the simulation test-bed the user can specify the threshold for the double talker algorithm. In our simulations we set the threshold to 9 dB under the assumption (which is true in our simulation) that the transhybrid loss is at least 9 dB.

Since the frequency domain algorithm is a block type algorithm, a slight modification has been made in the

double talker algorithm. In the frequency domain algorithm each block of N points is processed at the same time and the input data is overlapped so that in each block we get M new points when $M \leq N/2$. For each new input point the double talker algorithm works in the same mode as for the LMS algorithm, but the decision to adapt or not is based on M decisions at the same time. In the algorithm that we use, if there is more than one double talker decision in the M new points, a double talker flag is applied to the entire block.

In the double talker algorithm we have a double threshold option. One option is the threshold for the individual or point threshold as in the LMS algorithm. The second threshold is the number of double talker events in one block of points. In the simulation we ran, the first threshold was 9 dB, and the second threshold was more than one double talker decision in the M points. These thresholds were found experimentally.

5. Improved LPC Analyzer For Large Dynamic Range Signals

5.1 Introduction

In this section we discuss the possibility of using an Automatic Gain Control (AGC) in the full duplex channel with a vocoder in the loop. The main purpose of the AGC is to enhance the quality of the synthesized speech in low level signals. A low level signal can be caused either by line loss, which can be as high as 15-20 dB [3] for a long distance line, or by the speaker articulation.

Fig. 3 shows a block diagram pertaining to the simulation of a full-duplex telephone with a vocoder in the loop. This block diagram includes the A/D and D/A units, the echo canceller, the analyzer and synthesizer, and a digital simulation of the hybrid for the 2-wire to 4-wire conversion. The best place to introduce the AGC is before the A/D unit. Since the AGC must be an analog device which is highly nonlinear, its interaction with other components must be carefully examined--especially with the hybrid and the echo canceller. In Subsection 5.2 we discuss the problems that arise from such interaction.

Since the main conclusion of Subsection 5.2 is that the AGC performance becomes very complicated when interacting with other components, we chose in Subsection 5.3 to work

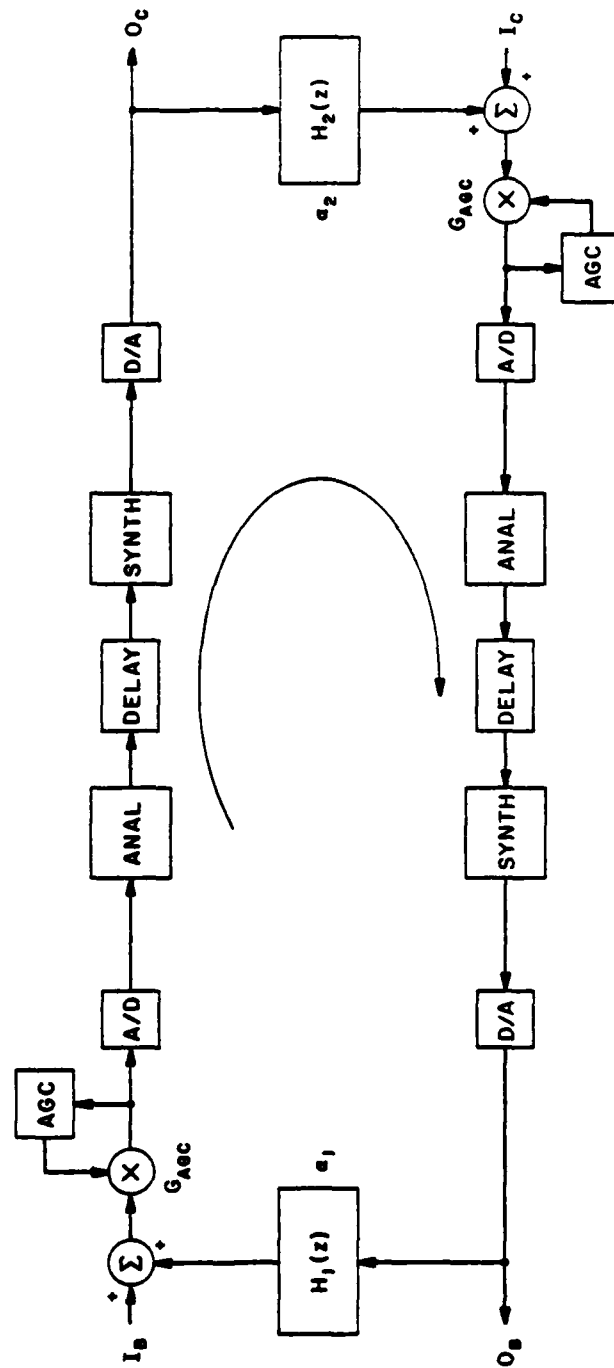


Fig. 19. A closed loop with block diagram of full duplex telephone network without echo-canceller

directly on the improvement of the LPC analyzer for a low level signal. In Section 5.4 we describe an experiment in which good quality telephone line speech has been achieved over a dynamic range of 30 dB, using the algorithm described in Section 5.3.

5.2 AGC Interfacing Problems

To analyze the effect of an AGC on the full-duplex system (Fig. 19) in an attempt to achieve better synthesis quality, some characteristic parameters of the AGC must first be defined. In the current case--an AGC followed by an A/D converter and an LPC analyzer--standard AGC parameters are:

- 1) Dynamic range of 30 to 40 dB: this parameter is necessary in offsetting the losses due to telephone lines which may be as high as 20 dB [3], and the naturally occurring energy variations due to articulation.

- 2) Fast response time of 10 to 30 msec: this is the time for the AGC to change from a high gain to a low gain. This fast response time is required to avoid overload upon precipitous high level input signals.

- 3) Slow release time of 0.5 to 1 sec: this is the time for the AGC to change from a low gain to a high gain. This slow response time is necessary to track the speech intonation.

The analysis of the system with such an AGC is very complex. From Fig. 19 we see that because of the hybrid leakage we have a closed feedback loop which may become unstable. The main difficulties in analyzing this closed loop system come from the nonlinearities of the LPC analyzer and the AGC, and the dynamic behavior of the echo canceller. Even more complexity is added from the adaptation of the echo canceller being frozen during simultaneous double talker conversation. To simplify the analysis we will check conditions for stability of the system without the echo canceller, and then analyze the effect of the AGC on the performance of the echo canceller. It can be shown that stability of the system without the echo canceller and stability of the echo canceller with the AGC assures the stability of the system under normal conditions.

5.2.1 Stability of the System Without Echo Canceller

To check conditions for stability without the echo canceller we may use the block diagram of Fig. 19. In this figure the transhybrid responses of the near-end and far-end talkers are represented respectively by $H_1(z)$ and $H_2(z)$. The block diagram includes the AGC, the A/D and D/A converters, the LPC analyzer/synthesizer, and the satellite delays. Since the LPC analyzer is nonlinear we cannot use

standard linear techniques to check stability of the system. However, we may use the Lp stability condition theorem [23] for the purpose (where sufficient conditions for stability require that the over-all closed loop gain be less than 1). Since the system is more complex when we include the echo canceller in the loop, this sufficient criterion assures the stability of the system. To simplify the analysis, we replace the transhybrid response $H_1(z)$ and $H_2(z)$ by a constant gain α_1 and α_2 for the hybrids, respectively, which in the worst case can be as high as -6 dB. If we use an AGC with dynamic range of 30 dB, unstability may occur since the over-all loop gain can be as high as 48 dB in the worst case. If G_{AGC} is the AGC gain in dB for both sides and α_1 and α_2 are the worst case leakage in dB for the hybrids, the overall closed loop gain will be

$$G_{CL} = 2G_{AGC} + \alpha_1 + \alpha_2 \quad (5.1)$$

To assure stability of the system we must add some loss in the closed loop system to compensate for the high AGC gain. Since the main function of the AGC is to maintain a high signal to quantization noise ratio at the input to the analyzer, the loss must be introduced after the analyzer along the transmitting path. In fact, the best place to insert this loss is after the synthesizer. The information about the amount of loss needed can be obtained from the AGC gain, and transmitted digitally through the channel along

with the information from the analyzer. Fig. 20 represents a block diagram which includes an analog controlled loss after the synthesizer. An analog instead of digital loss is suggested to maintain low quantization noise for low level signals at the synthesizer output. An accuracy of 3 dB in the controlled loss is enough to assure stability since a 6 dB minimum loss comes from each hybrid, yielding a singing margin of 6 dB for the system in the worst case (See NSC Note 139 for definitions) [4]. Therefore, an additional 4 bits per frame are needed to transfer the AGC information to the controlled loss through the channel. Since the information from the LPC is transmitted for every frame, the AGC gain information must also be sent for every frame. Therefore, the change in the AGC gain must occur on frame boundaries only.

A partial conclusion from the discussion so far is that an AGC can be introduced as shown in Fig. 20 under the following conditions:

1. A gain change in the AGC, can occur at frame boundaries only.
2. A 4 bits quantization of the AGC gain must be transmitted through the channel for an analog controlled loss.

In the next section we analyze the effect of the AGC on the echo canceller.

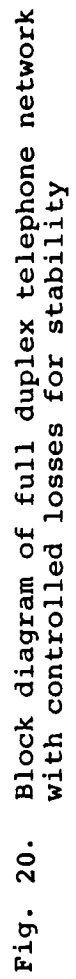


Fig. 20. Block diagram of full duplex telephone network with controlled losses for stability

5.2.2 AGC--Echo Cancellation Interaction

From the last section we saw that an AGC, with a fast response time of 22 msec (frame size), can be incorporated into the system and can have a dynamic range of 40 dB without causing stability problems. The question then is how such an AGC affects the performance of the echo canceller. In Fig. 21 we show a block diagram which includes the hybrid, the AGC, A/D and the echo canceller with a double talker detector. To achieve good performance from the echo canceller, the adaptive filter must compensate for the transhybrid response plus the instantaneous gain of the AGC. Since we have the information about the instantaneous gain of the AGC, this information can be incorporated into the adaptive filter. The same gain information can be used by the double talker algorithm to compensate for the gain introduced by the AGC.

In the ideal case, where the compensation in the two cases is exact and under the assumption that the hybrid is linear, there is no interaction problem and, as a result, the echo canceller will have the same performance with or without the AGC. Practically, however, we have to be concerned about how an error in the compensation and non-linearity in the hybrid will affect the performance of the echo canceller.

A straightforward calculation of the effect of a gain

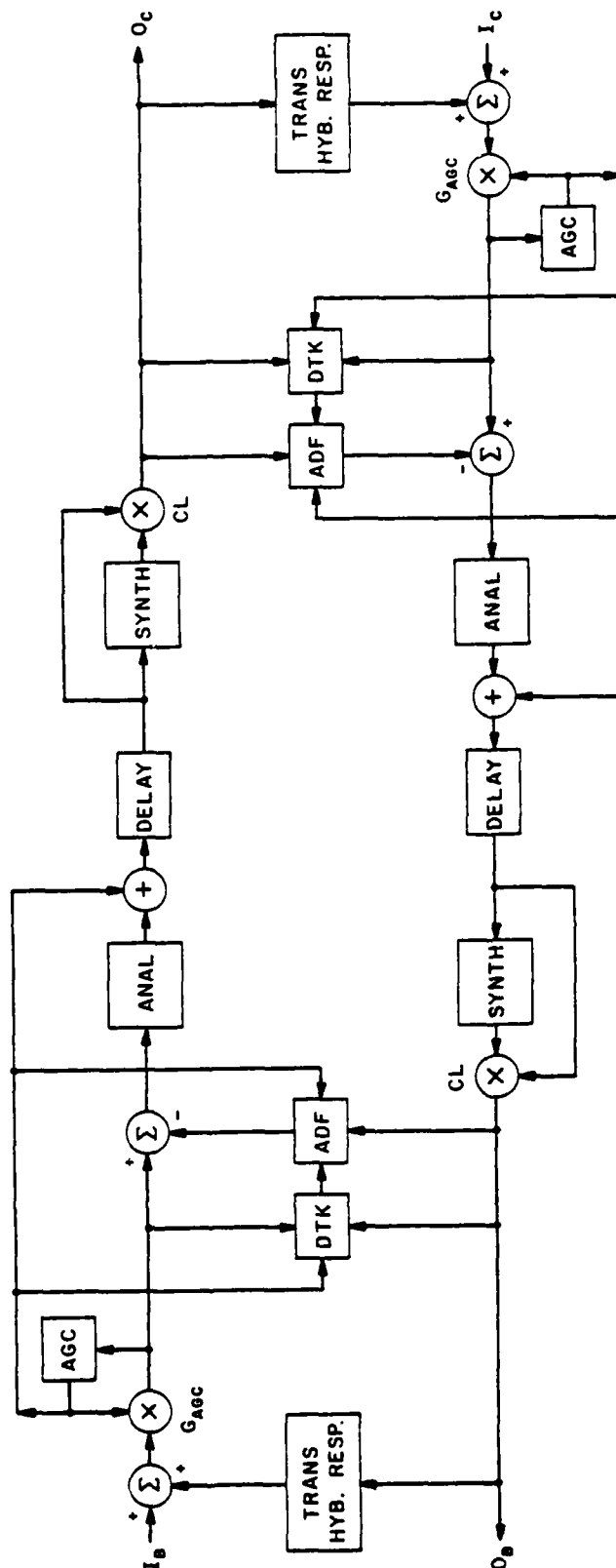


Fig. 21. Proposed full duplex block diagram with AGC compensations

error on echo cancelling performance will give us the following result: an error of 1% in gain will limit the echo reduction to a maximum of 40 dB. During the adaptation phase the echo canceller will try to compensate for this error each time the gain is changed, but during a double talker situation the adaptation is frozen and the maximum echo reduction in this case is limited to 40 dB.

We assumed so far that the hybrid is linear and can be modeled by a linear filter. From measurements of the transhybrid response [13], we found that a linear filter approximates the transhybrid response only up to 25-28 dB; this means that with a linear adaptive filter the maximum echo reduction can be 25 to 28 dB. From those measurements we found also that a second-order Volterra series can achieve 8 dB more echo reduction [13]. In a system without AGC the maximum level of the echo will be given by the dB sum of the transhybrid loss and the maximum attainable echo reduction. In the worst case the transhybrid loss is 6 dB, and if we add to it the maximum echo reduction of 28 dB we see that the maximum level of the echo becomes -34 dB. When we introduce an AGC in the loop, the gain of the AGC must be added also. If, for example, the gain is 20 dB, the maximum level of the echo will be -14 dB, which is intolerable for long distance calls.

A conclusion from the above discussion is that a complex AGC with special specifications can solve the

oscillation problems and achieve a low quantization noise for low level signals at the LPC analyzer input. However, such an AGC will reduce the echo-reduction achieved by the echo canceller algorithm to an unacceptable level. This loss in the echo-reduction performance is due mainly to the non-linearity of the hybrid.

In the next section we present a different approach to solving the same problem by directly improving the existing LPC analyzer to perform better for larger dynamic ranges.

5.3 Pitch and Voicing Algorithm Improvement

Pitch and voicing estimation is probably one of the most difficult and challenging problems in speech analysis for narrow band voice coding. It is even more difficult to design an algorithm that will work well for speech signals which have a very wide dynamic range. Although we have shown in a previous note [11] that the cepstrally based pitch and voicing estimation method gives accurate pitch and voicing analysis results and is particularly robust in a noisy environment, substantial degradation in the voicing decision was recently found when it was applied to speech signals of extremely low level with peak amplitude of the order of 5-6 bits. In this section we try to improve the algorithm used for dealing with extremely low level signals.

In this section we describe the modifications made to

the original algorithm [11] and report in the next section experimental results of applying the modified algorithm to speech segments with abrupt changes of over 20 dB in signal level. Significant improvement in performance will be shown and its effect in Wideband Integrated Network Communication will be discussed.

5.3.1 Modifications

As reported in [11], four parameters are used in making the voicing decision--the first reflection coefficient, the energy parameter, the zero crossing rate, and the cepstral peak value. These parameters and the voicing decision logic that uses these parameters are the focus of the modifications which are listed below:

A) The first reflection coefficient (K_1)

The first reflection coefficient appears to be the parameter least affected by inadequate signal level. However, it is desirable to use in voicing decision a K_1 that is obtained independently of analysis conditions such as the pre-emphasis factor. Since K_1 provides information on the first order spectral shape, we prefer to use a K_1 obtained from a signal that is constantly pre-emphasized by a factor of 0.9. Such a factor will lead to a rough voicing indication, in general, a positive K_1 for unvoiced frame and a

negative K_1 for voiced frame. Unfortunately, such a pre-emphasis factor may require extra signal processing since the pre-emphasis factor in a general vocoder analyzer may be differently specified. In order to avoid introducing extra complexity in computing the 0.9 pre-emphasized K_1 we use the following approximations. Let $x(n)$ and $y(n)$ be the input sequence and the pre-emphasized sequence with a factor of p , respectively. That is,

$$y(n) = x(n) - p x(n-1) \quad (5.2)$$

Expressing the first two autocorrelation terms of $y(n)$ in terms of the autocorrelation of $x(n)$, we have

$$\begin{aligned} r_y(0) &= \sum y^2(n) \\ &= \sum x^2(n) - 2p \sum x(n) x(n-1) + p^2 \sum x^2(n-1) \\ &= r_x(0) - 2pr_x(1) + p^2 r_x(0) = \\ &= (1+p^2) r_x(0) - 2pr_x(1) \end{aligned} \quad (5.3)$$

$$\begin{aligned} \text{and } r_y(1) &= \sum y(n)y(n+1) \\ &= (1+p^2)r_x(1) - p[r_x(2) + r_x(0)] \end{aligned} \quad (5.4)$$

As a result, the first reflection coefficient K_1 corresponding to sequence $y(n)$ is

$$K_1 = - \frac{r_y(1)}{r_y(0)} = \frac{p[r_x(2) + r_x(0)] - (1+p^2)r_x(1)}{(1+p^2)r_x(0) - 2pr_x(1)} \quad (5.5)$$

In this case where $P = 0.9$.

$$K_1 = \frac{0.9 [r_x(2) + r_x(0)] - 1.81 r_x(1)}{1.81 r_x(0) - 1.8 r_x(1)} \quad (5.6)$$

This equation thus eliminates the need for an extra pre-emphasis operation, and the K_1 will be approximately the first reflection coefficient corresponding to a constantly pre-emphasized (with $P = 0.9$) data sequence for the pitch and voicing algorithm.

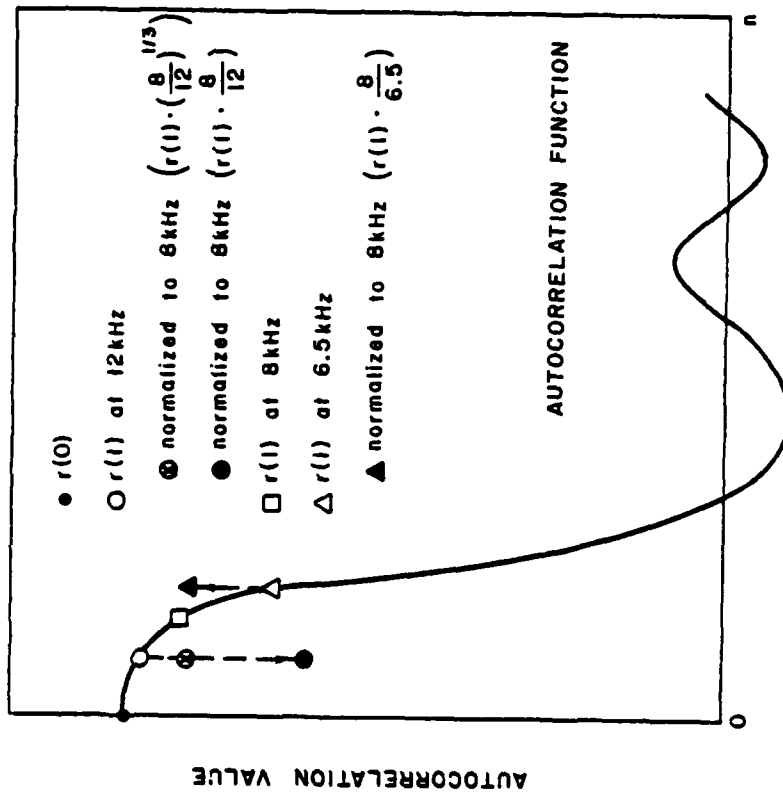
Another modification to K_1 is in the normalization of the parameter to a corresponding one with 8 KHz sampling frequency. In [11], the normalization factor is defined as

$$\eta = (8000/f_s) \text{ for all } f_s \quad (5.7)$$

where f_s is the sampling frequency (in Hz) of the data to be analyzed. Such a normalization factor is generally appropriate for $f_s > 8$ KHz due to the shape of the autocorrelation function as illustrated in Fig. 22. However, it is too far off for $f \approx 8$ KHz., resulting in more voiced to unvoiced errors due to over-reduced K_1 score. We have found from experiment that a normalization factor of

$$\eta = (8000/f_s)^{1/3} \quad (5.8)$$

is more suitable for $f_s > 8$ KHz. Such modification significantly improved the voicing decision for $f_s > 8$

Fig. 22. Normalization of $r(1)$

KHz.

B) The Energy Parameter

The energy parameter should be carefully defined particularly when the input level is low. More experiments have found that the signal energy is more appropriate in voicing decision than the residual energy. Therefore, instead of the residual energy which was used in [32], we now employ the signal energy. Note that the energy parameter is a complicated function of the energy term as discussed in [11]. The simple change from the residual energy to the signal energy has more complication than is implied in the terms (refer to [11] for details).

As the signal energy is always higher than the residual energy, the two terms RMSUV and RMSAVE, which represent the smoothed unvoiced energy and the overall energy contours, require a higher initial value. We have found that a value of 512 (increased from the previous 256) gives good results.

C) The Zero Crossing Rate

In [11], an integer bias term was used in the zero crossing count. The bias term alternates its sign for successive samples to avoid a very small zero crossing rate when the signal is almost a constant. The term was defined as

$$b = \text{INT}(\text{RMSUV}/128) + 4$$

where INT and RMSUV denote the truncation to integer operation and the smoothed unvoiced energy respectively.

Apparently, the integer operation and the constant term "4" will cause problems in obtaining a reliable zero crossing count when the signal level is low. In particular, the constant term will lead to an excessive zero crossing rate for low level signals resulting in voiced to unvoiced errors. Deleting this constant term in turn may yield a zero bias, due to the integer operation, that will result in a very small zero crossing rate and an unvoiced to voiced error when the signal is low.

The following modification is thus made:

$$b = \text{RMSUV}/128 \quad (5.9)$$

By deleting the constant term and resorting to floating point operation, a much more reliable zero crossing count is obtained in dealing with low level signals.

D) Voicing Decision Logic

Only minor changes in the voicing decision logic were made. The modification involves the non-linear score for an extremely high or extremely low cepstral peak, and is listed as follows:

i) $\text{ICX} = 250$ (was 180) when $\text{ICPT} \geq 60$ and

ii) $\text{ICX} = -200$ (was -400) when $\text{ICPT} \leq 10$ (was 19)

where ICPT and ICX denote the normalized cepstral peak

value and the voicing score due to cepstral peak respectively. It can be seen that the modification emphasizes the voicing for high cepstral peaks and softens the unvoicing override when the cepstral peak is low, a case that may occur for a voiced frame when the signal is low.

As will be demonstrated in the next section these modifications greatly improve the voicing accuracy. Also, the pitch accuracy at the same time is well maintained as compared to the same signal of higher level.

5.4 Experimental Results

To check the performances of the existing LPC analyzer on large dynamic range signals we set up the following experiment. A speech sample 60 sec long was digitized by a 12-bit linear A/D and then analyzed and synthesized for reference. Then the original speech data was divided in three equal parts of 20 sec each. The first section remained the same, the second section was divided by 8 and the third section by 32. In this way we formed test speech with more than 30 dB dynamic range. This test speech has been analyzed and synthesized by both the existing LPC algorithm and the improved algorithm described in the last section. The synthesized speech from the existing LPC

analyzer was highly distorted mainly because of errors in voice/unvoice decisions. The improved pitch algorithm achieves a fairly good synthesized speech without errors in voice/unvoice decisions and the quality of the synthesized speech was comparable to the original synthesis.

To illustrate the improvement in the voice/unvoice detection algorithm, we created a new test speech segment. This new test segment includes 60 speech frames. The first 20 frames were chosen from the original speech data, the next 20 frames are the same first 20 frames but divided by 8, and the last 20 frames are the same first 20 frames divided by 32. In the upper part of Fig. 23 we have the display of this new test segment, and in the lower part we show the synchronized pitch and voicing output using the existing pitch algorithm. A number of voice/unvoice detection errors at low level speech signal can be observed. The corresponding pitch and voicing output of the test segment using the improved algorithm is shown in the lower part of Fig. 24. We can see that with the improved algorithm no such errors occur.

To illustrate the effect of quantization noise on the reflection coefficients we present Fig. 25. Fig. 25a presents the smoothed LPC spectrum of 10 consecutive frames of the original speech, and Fig. 25b presents the smoothed LPC spectrum for the same frames with 18 dB lower signal to quantization noise ratio, and Fig. 25c presents the smooth

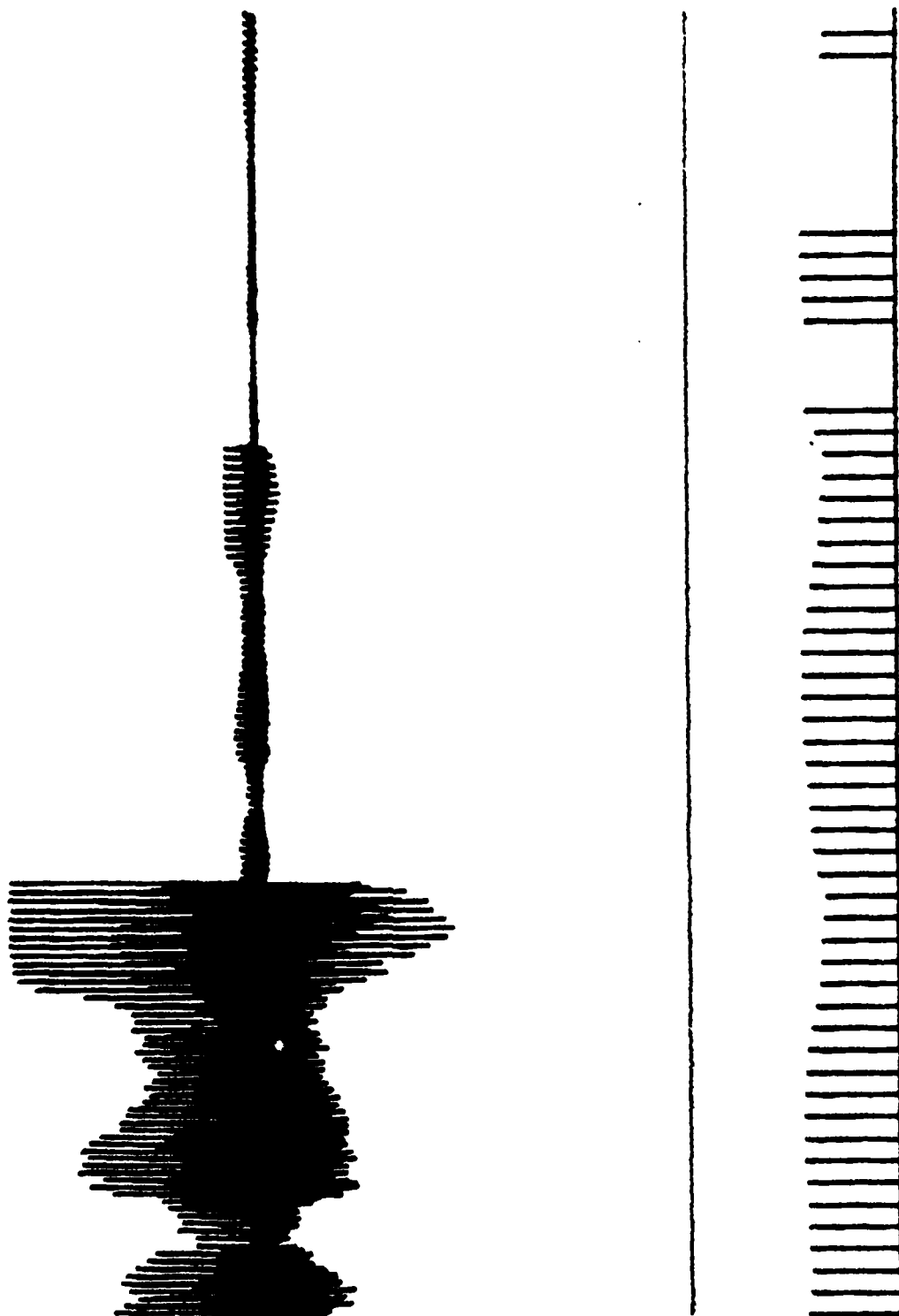


Fig. 23. Input test signal and pitch output with old pitch analyzer

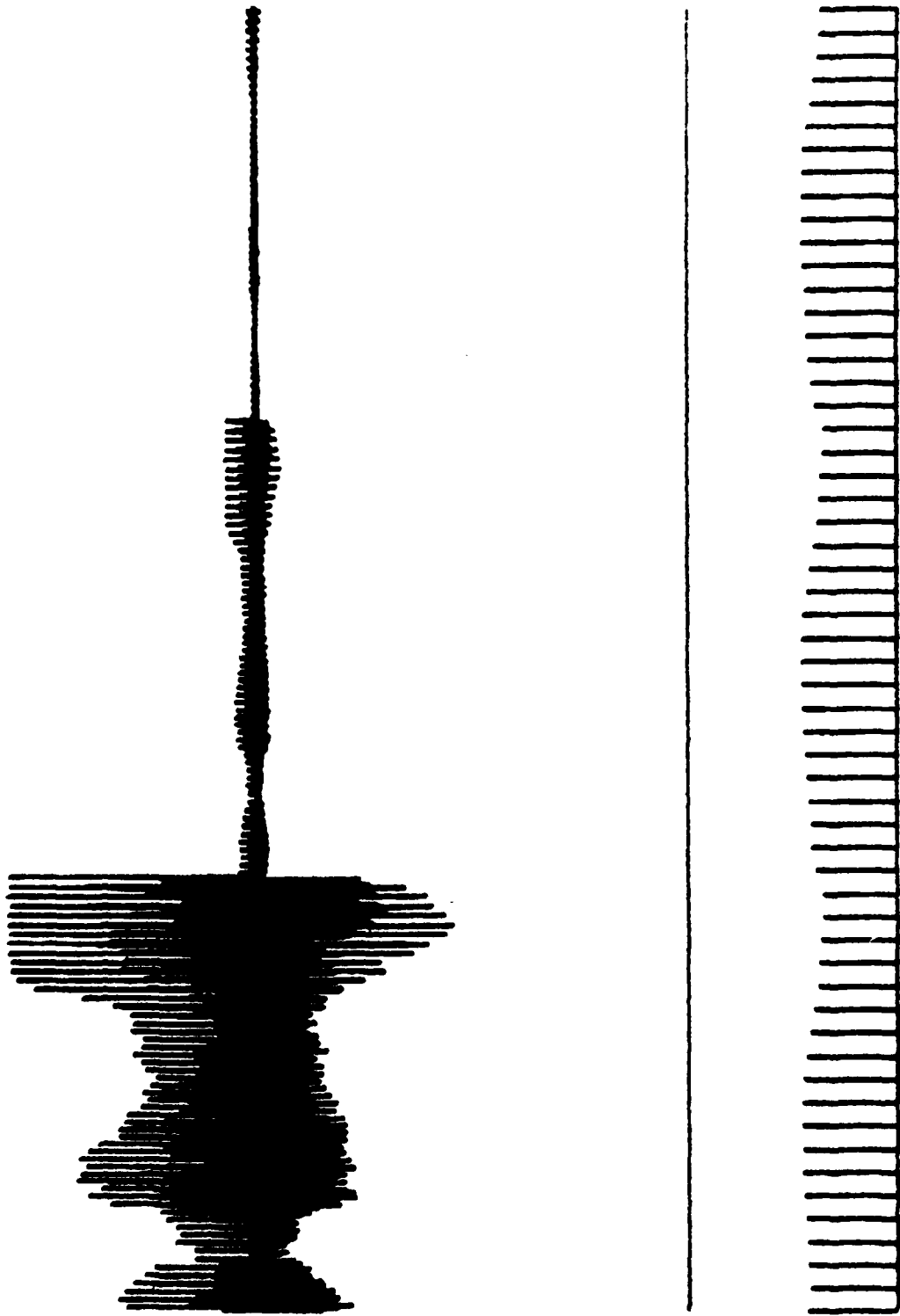


Fig. 24. Input test signal and pitch output with improved pitch analyzer

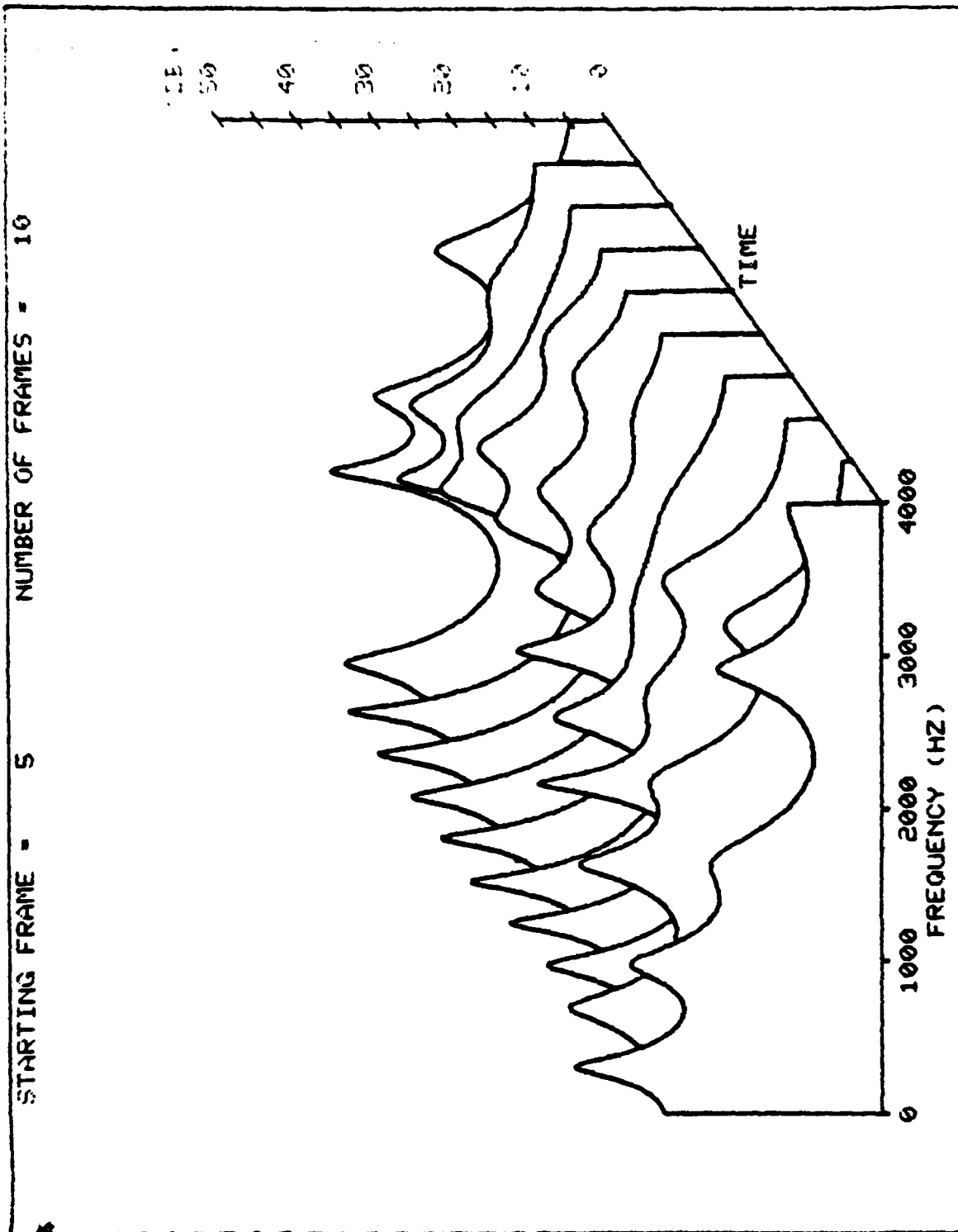


Fig. 25. LPC smoothed spectrum for input test signal at different input levels

a. 0 db level

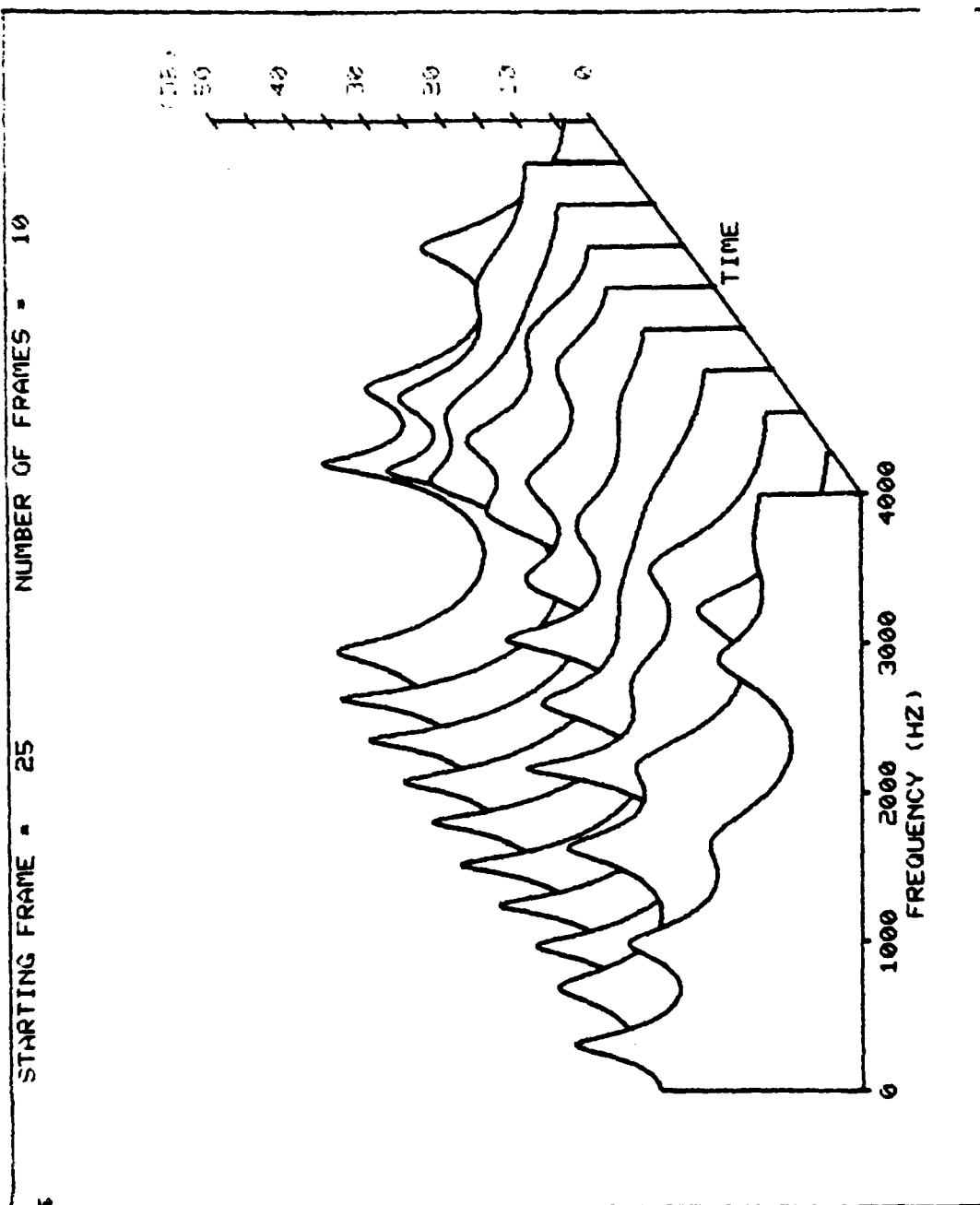


FIG. 26. -18 db level

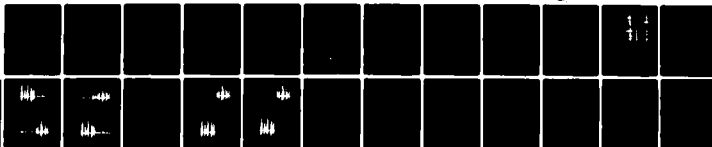
AD-A112 604

SIGNAL TECHNOLOGY INC SANTA BARBARA CA F/O 17/2
 ALGORITHM DEVELOPMENT OF A TELEPHONE SIGNAL CONDITIONER FOR THE--ETC(U)
 DEC 81 A H GRAY, D MANSOUR N00014-76-C-0214

UNCLASSIFIED

NL

242
 242 (A)



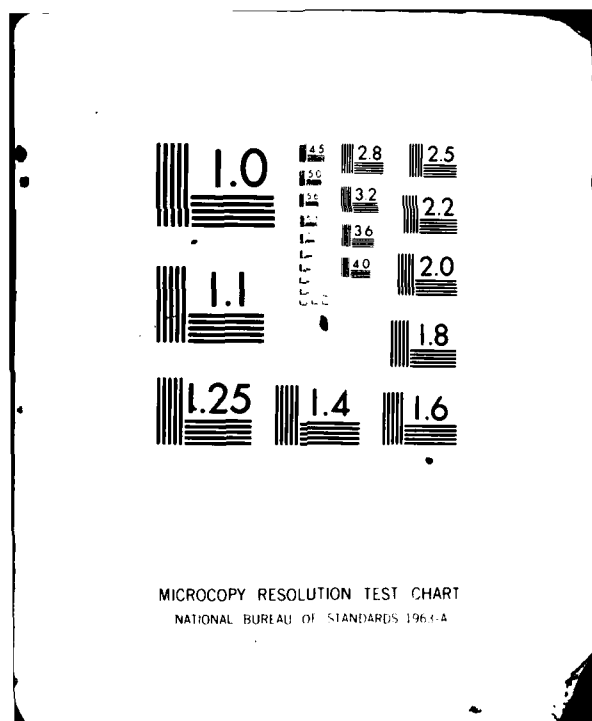
END

DATE

FILED

82

DTIC



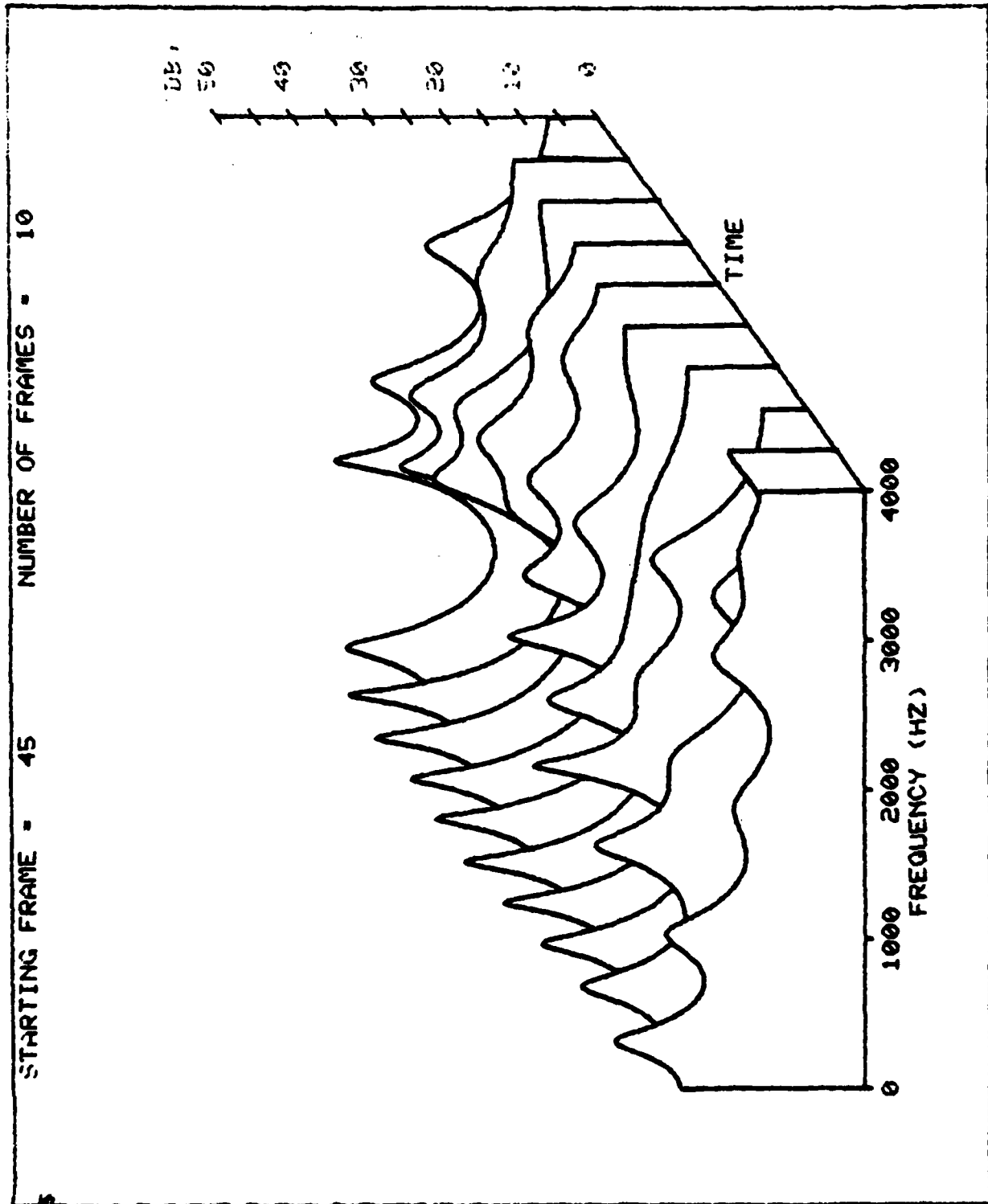


Fig. 27. -30 db level

power spectrum for the same frames with 30 dB lower signal to quantization noise ratio. We can notice from the results that the distortion introduced by the quantization noise is fairly low. This confirms our results from informal listening that the synthesized speech with the improved pitch algorithm is fairly good for a large dynamic range in speech input.

5.5 Conclusion

Our main conclusion from the study reported in this section is that an improved analyzer, as introduced in this note, solves the dynamic range problem at the LPC input analyzer. This solution is much more efficient than an AGC from both complexity and performance points of view.

Although an AGC looks intuitively attractive, a careful check of its interaction with the entire system leads to the conclusion that even a highly complex AGC still reduces the performance of the echo cancelling algorithm.

6. Experimental Results

6.1 Introduction

The main purpose in developing the test-bed simulation was to check the efficiency of the algorithms described so far in the context of the entire full-duplex network. During the entire period of the project a lot of experiments were run. In fact, every algorithm inserted in the test-bed simulation was first tested by a special test program, and the best parameters for the specific algorithm were chosen. In this section we report those final experiments that were done to check the interaction between a specific algorithm and the full-duplex system, under real time conditions. Since the decisions on algorithm parameters and their efficiency are based on the subjective speech quality output of the system, the description of the results will be more qualitative.

The organization of this section is as follows. First we describe the options available in the test-bed simulation. Different network configurations can be designed with these options. Then we present a number of experiments done to check the following main points:

1. Echo effect, on speech perception, as a function of the delay length.
2. Effect of LPC vocoder in a full-duplex network without echo canceller.
3. Performances of echo canceller algorithms without vocoder in the loop.
4. Study of echo canceller algorithms with vocoder in the loop.
5. Performances of echo canceller algorithms with time-variant transhybrid response.

6.2 Test-bed Simulation Options

The test-bed simulation program is very flexible and by means of "software switches" allows the user to choose different configurations and different parameters for various experiments.

The main options are:

- with or without vocoder in the loop
- with or without double-talker algorithm
- with or without echo cancelling algorithm

By means of "software switches", we can choose one of four different adaptive filter algorithms, we can choose different hybrid responses or the kind of filter (IIR or FIR

filter as the transhybrid response). We can also make experiments with one or both talkers active at the same time.

Each algorithm can be controlled by a different parameter. Part of the parameters are controlled through the common array from the ILS system, other parameters are introduced directly through the simulation program. By use of these parameters the user can change the conditions or thresholds of different algorithms.

6.3 Experimental Results

6.3.1 Experiment 1: Echo effect on speech perception as a function of the delay length.

In this experiment we choose the "software switches" in the test bed simulation so that we got a very simple model which included only the programmable delay line and a simple transhybrid loss. The transhybrid responses chosen in this experiment were simple gain factors β_1 and β_2 , as in Figure 26.

We ran the digitized sentences from two speakers, collected in the same way as described in section 2.3.1. We ran the same data with four different parameters: two different parameters for the transhybrid response, ($\beta_1 = \beta_2 = -5\text{DB}$ and -10 DB), and two different lengths for the delay ($L_1 = L_2 = 100\text{ msec}$ and 400 msec).

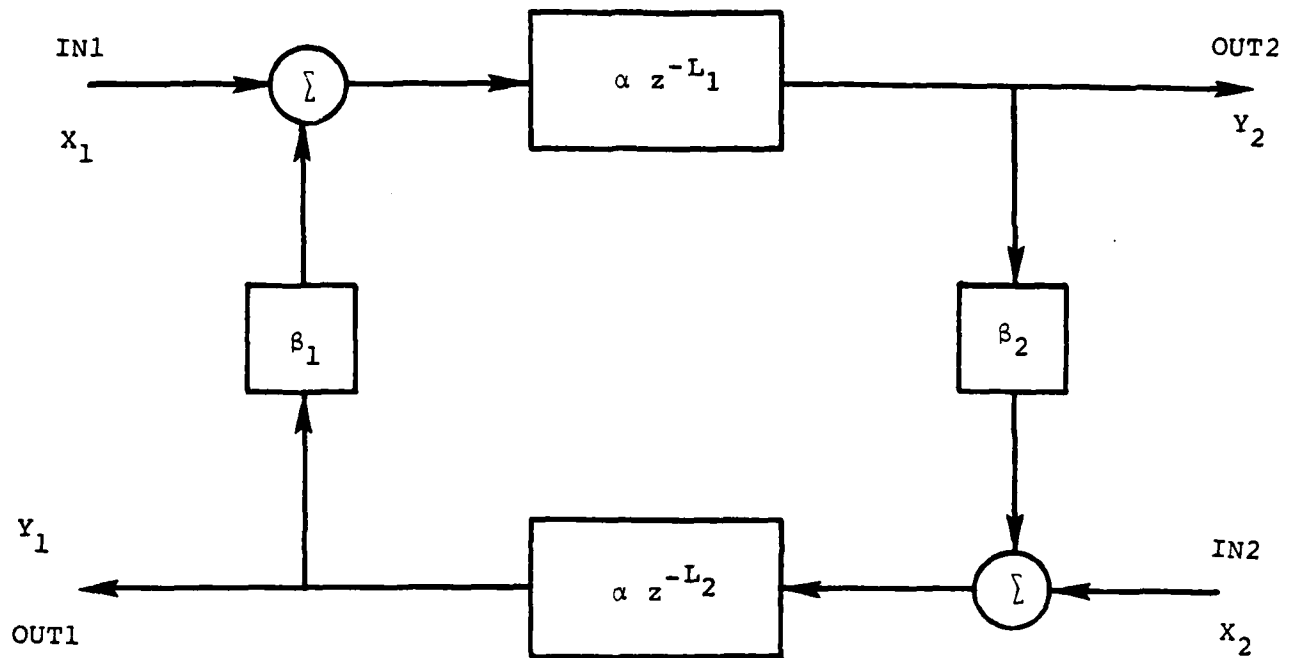


Fig. 26. Simplified full-duplex telephone line model.

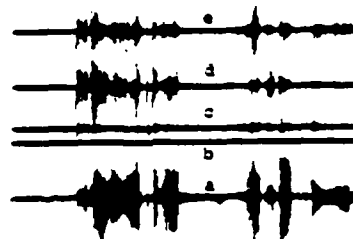


Fig. 27 Output: (a) without echo canceller, (b) with transhybrid response equal to zero (ideal), (c) with frequency domain LMS algorithm, (d) with Widrow LMS algorithm, and (e) with gradient FIR lattice algorithm. Identical scale for b-e, but scale of a is three times larger than the scale of b-e.

By listening to the outputs, with all 4 different combinations for β_i and L_i , it was perceptually apparent that the effect of the echo was increased as L_i increased from 100 msec to 400 msec; of course with higher loss for the hybrid the effect of the echo decreased. However, with the higher loss and longer delay, the perceptual effect of the echo still increased. This result confirmed results reported in the literature [3] on real satellite communication telephone line. An interesting point to notice is that even without the psychological effect of the long delay on the talker's conversation, longer delay increased the effect of the echo.

6.3.2 Experiment 2: Effect of LPC vocoder in the full duplex network without echo canceller.

In this experiment the "software switches" were chosen so that the full-duplex system includes a real simulation of the transhybrid response represented by 64 points F.I.R. filter, the LPC analyzers and synthesizers, and a transmission delay time of 300 msec.

The digitized sentences from two speakers collected from real telephone lines were used as input to the test-bed simulation program. The sentences were specially chosen such that a number of double talker situations occur. The average transhybrid loss used in this simulation was 9 dB. We ran the data with and without an LPC vocoder in the loop.

The main result from this test was that, with the same

transhybrid response for both cases, the effect of the echo was much more annoying with the vocoder than without the vocoder. During double talkers situations, the distortions introduced by the LPC analyzer were so high that intelligibility was lost.

The conclusion from this test was that with a vocoder in the loop a faster convergence rate is needed for the echo cancelling algorithm.

6.3.3 Experiment 3: Performances of echo cancellers without a vocoder.

The purpose of this experiment was to check the performances of three adaptive filter algorithms--the LMS algorithm, the gradient lattice algorithm and the unconstrained frequency domain algorithm--as echo cancellers. The test-bed simulation "software switches" were chosen so that the full-duplex system would include the transhybrid response, a transmission delay line of 300 msec and one of the echo canceller algorithms mentioned above. In this experiment the double talker algorithm was active. With each adaptive algorithm the test-bed simulation program selected the appropriate double talker algorithm. The double talker parameters, in this experiment, were selected to get the best performance.

A digital conversation, between two speakers, was collected over real telephone lines and run through the test-bed simulation with the three different algorithms.

The conversation was planned so that the far-end speaker was silent during the beginning part of the conversation. The output at the near-end side at the beginning of the conversation is given in Figure 27, for different conditions. Since the far-end speaker was silent at the beginning of the conversation in Figure 27 we observe only the returned echo at the near-end side. In Figure 27c we observe the faster convergence of the unconstrained frequency domain algorithm compared to the LMS Figure 27d and gradient lattice algorithm Figure 27e. In this case the UFLMS achieved 43 dB echo cancellation after 250 msec compared to only 30 dB for the LMS algorithm after 250 msec under the same conditions. However, both algorithms converge to the same final echo cancellation of 50 dB.

With the lattice algorithm, the echo cancellation obtained was 33 dB in 250 msec but the final echo cancellation was 40 dB. The poor performance of the gradient lattice algorithm can be seen in Figure 27d. Faster convergence for the gradient lattice algorithm can be achieved with higher value for the convergence constants but serious distortion, such as transient spikes, was introduced. The poor performance of the gradient lattice technique is due to the non-stationary behavior of speech. This conclusion is confirmed on large amounts of speech data. For a short segment of speech fast convergence can be achieved with the gradient lattice algorithm, but for long

speech data, because of the speech non-stationarity, the convergence constants must be chosen small enough to avoid unstable behavior, or else the performance of the algorithm may become very poor.

A conclusion from this experiment is that the UFLMS achieves best performance in both fast convergence and final amount of echo reduction after convergence. The lattice algorithm has faster convergence than the LMS algorithm but because of speech non-stationarity the algorithm does not converge to the optimal solution.

6.3.4 Experiment 4: Performance of echo canceller with an LPC vocoder in the loop.

The main purpose of this experiment was to check the interaction between the LPC vocoder and the echo cancelling algorithm. In this experiment we added the LPC analyzer and synthesizer to the test-bed simulation set-up which was used in the last experiment.

The last experiment has been repeated with the vocoder in the loop. The results are given in Figure 28. Figure 28a shows the near-end output in a free echo condition; in fact, since there is no echo this is the information sent by the far-end speaker, and we see that the far-end talker is silent at the beginning of the conversation. Figure 28b shows the output at the near-end side when the system includes the transhybrid response but not the echo canceller. Since the far-end talker is silent

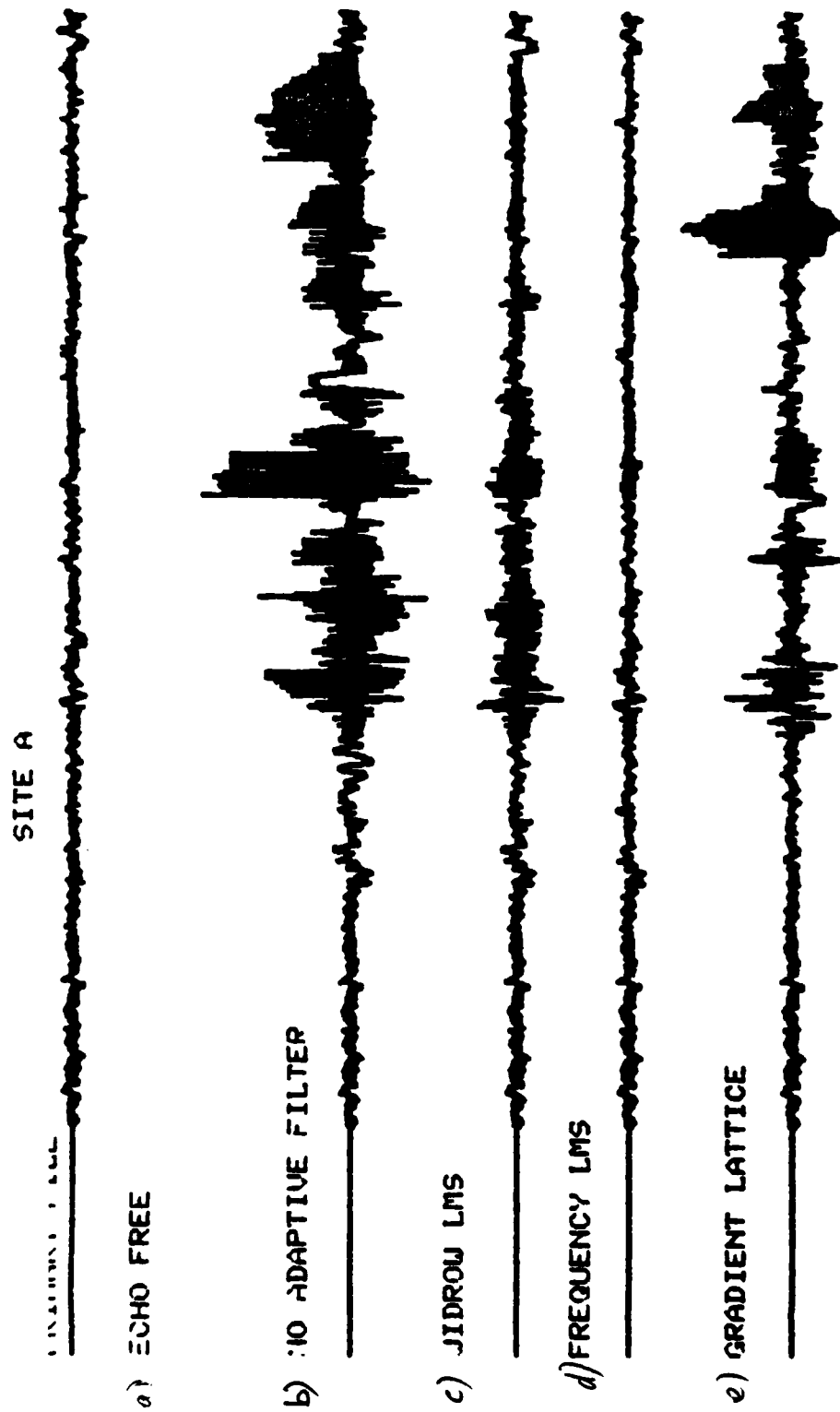


Fig. 28. Output of a full duplex channel with vocoder in the loop (site A).

at the beginning of the conversation, Figure 29b presents the returned echo at the near-end side without the echo canceller. Figure 28c presents the near-end output when the LMS algorithm was used as the echo canceller. Figure 28d presents the near-end output when the frequency domain algorithm was used as the echo canceller; from this result we observe the faster convergence of the UFLMS compared to the Widrow algorithm. Figure 28e presents the near-end output when the gradient lattice algorithm was used as the echo canceller algorithm. From Figure 28e we observe the poor performance of the gradient lattice algorithm. The interesting result was that with vocoder in the loop the gradient lattice performs worse than without the vocoder in the loop. Without the vocoder in the loop we could find a convergent constant such that for a conversation of 30 sec the adaptive filter was stable. With the vocoder in the loop we could not find such convergence constants. With every convergence constant some instability occurs in one place or another. Only with very low convergence constants was the adaptive filter stable; but then we had very poor echo cancelling. It appears that the main reason for this behavior of the gradient lattice algorithm is that the LPC synthesizer output is much more nonstationary than the original speech; the abrupt change of the prediction coefficient every frame introduced spikes at the gradient lattice output since the lattice algorithm is designed for

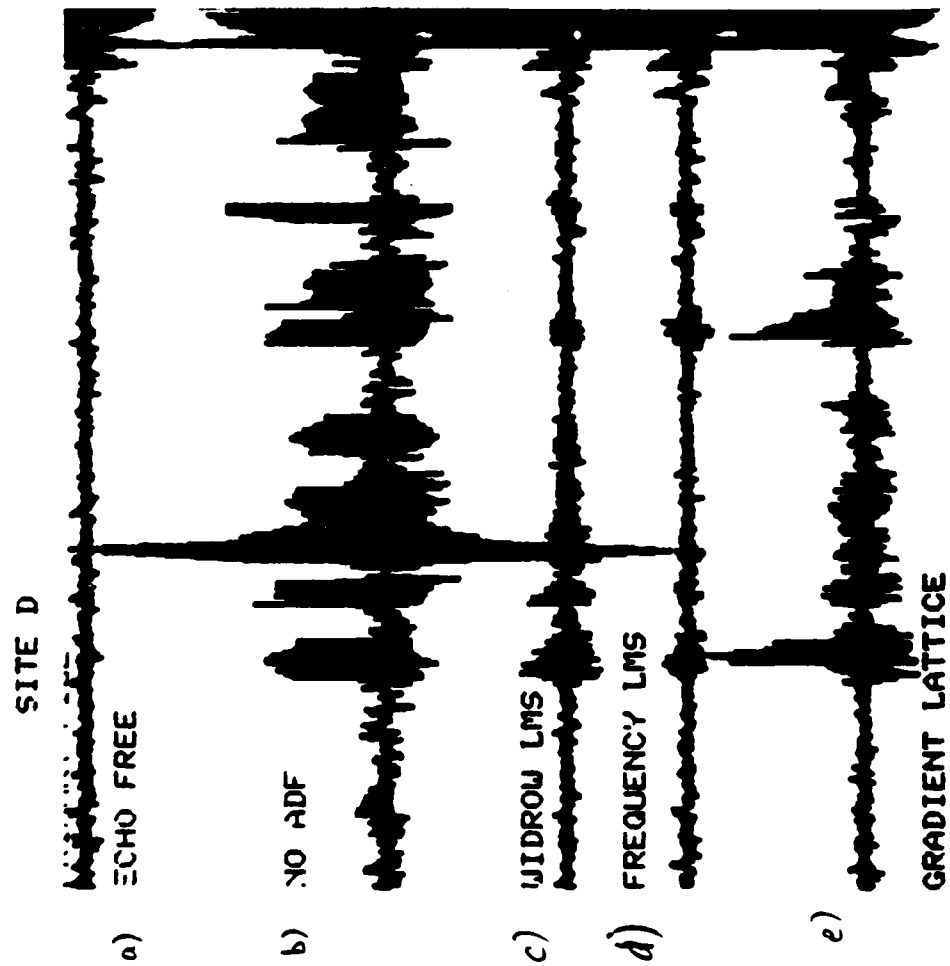


Fig. 29. Output of a full duplex channel with vocoder in the loop (site D).

stationary input.

In Figures 29a to 29e we have the same output at the far-end output, we see that the results at the far-end output behaves the same. The main result from this experiment was that with the vocoder in the loop, the frequency domain algorithm has better performance than the LMS algorithm and that the lattice algorithm cannot be used at all in our application.

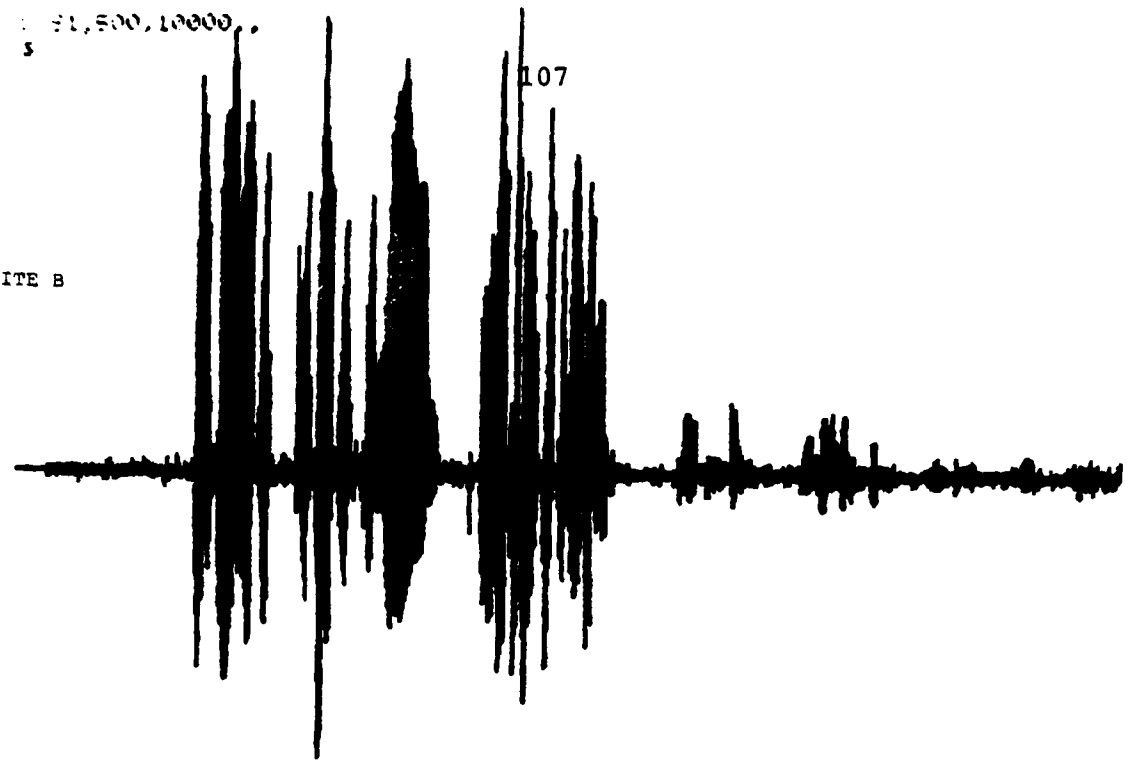
6.3.5 Experiment 5: Performance of echo cancellers with time-variant transhybrid response.

In the experiments presented so far the transhybrid response was fixed during all the experiments. As explained earlier the transhybrid response library includes 14 different loading conditions. In the test-bed simulation program there is a "software switch" for time variant transhybrid response. By activating this switch the user can specify a time interval T so that every T seconds the transhybrid response is changed successively from the library.

In this experiment we use the time variable switch with $T = 1$ sec and run a conversation of 50 sec through the test-bed simulation with the LMS and the UFLMS algorithms. The results of this experiment are given in Figure 30-33.

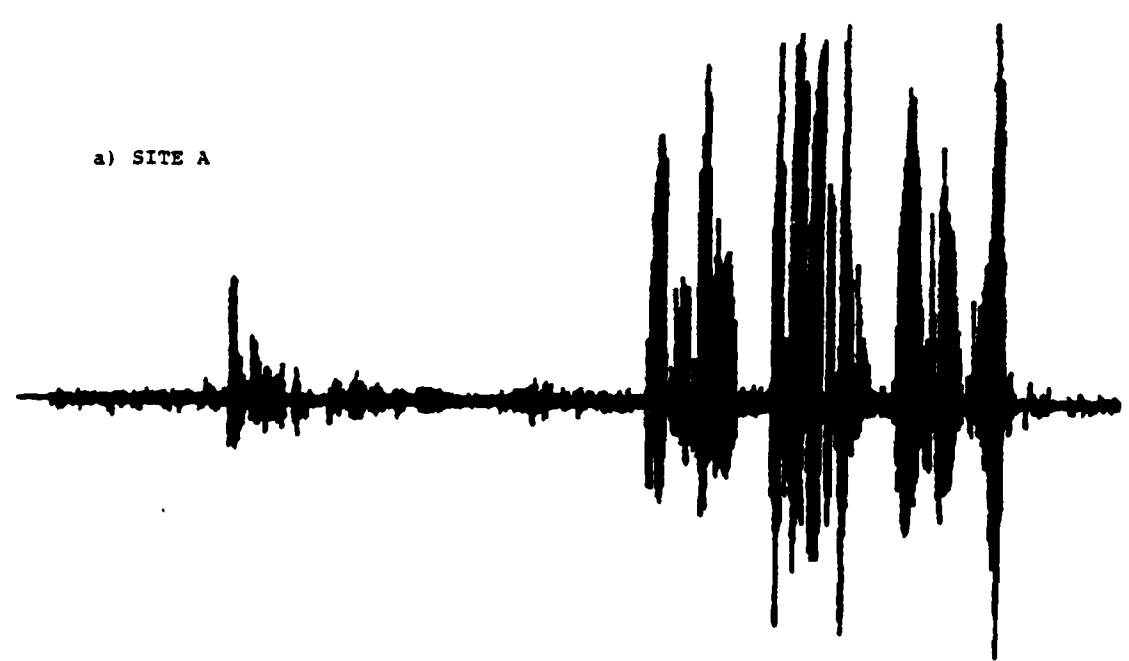
Figure 30 and 31 show the near-end and far-end outputs with the LMS algorithm. Figure 30 presents the results with constant transhybrid response, Figure 31 presents the

51,500, 10000
5
107
b) SITE B



This seismic waveform for Site B shows a baseline with low-level noise. A significant seismic event occurs between approximately 20% and 40% of the time axis, characterized by a dense cluster of high-amplitude, high-frequency oscillations. The signal returns to the baseline level after this event.

51,500, 10000
a) SITE A



This seismic waveform for Site A shows a baseline with low-level noise. A seismic event begins around 60% of the time axis, featuring several distinct, high-amplitude peaks. The signal returns to the baseline level after the event concludes.

Fig. 30. Output - (LMS) algorithm with constant hybrid response.

FIG 1351

0 250 500 10000

108

SITE A

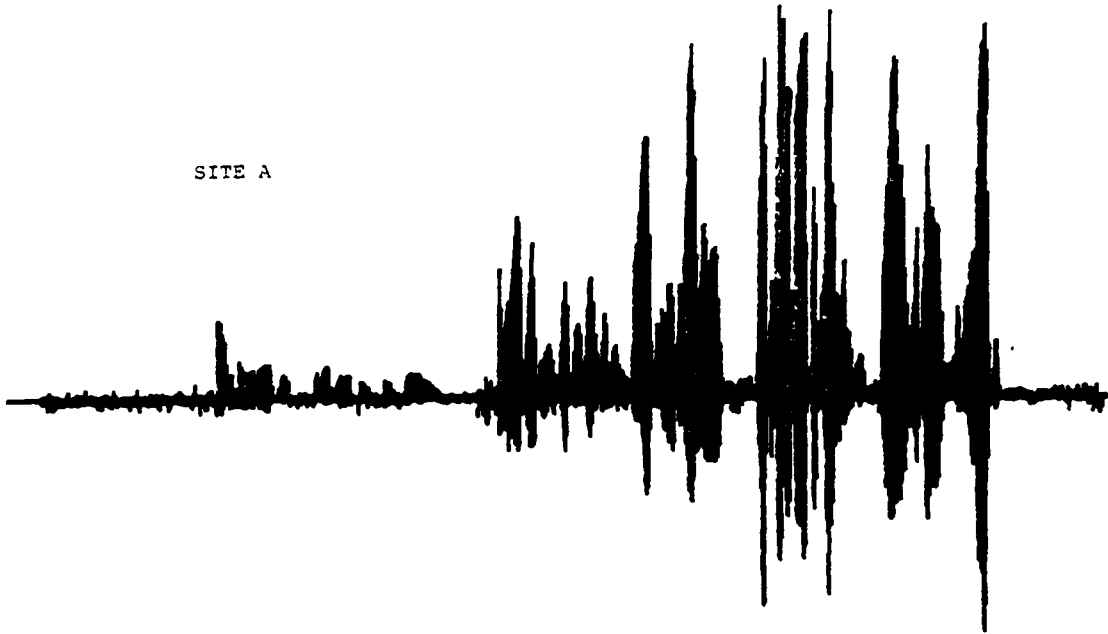


FIG 1362

0 250 500 10000

SITE D

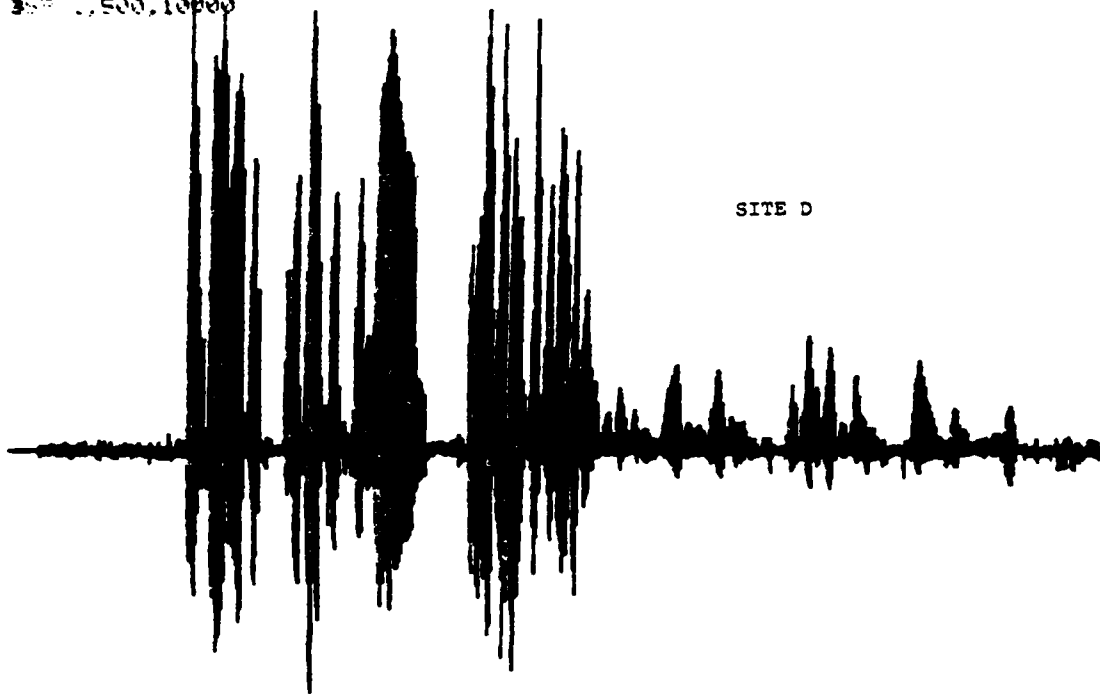


Fig. 31. Output at site A and D - LMS algorithm with dynamic transhybrid response.
(Changes every 1 sec--the display is 10 sec)

results with the dynamic transhybrid. It is clear from those results that for such conditions the LMS algorithm did not achieve good echo reduction. From those figures we can observe the high echo in Figure 31 compared to the low echo in Figure 30 after convergence.

Those results were obtained with a convergence factor of $\mu = 0.1 \times 10^{-7}$, which was the highest convergence constant possible without a stability problem. An interesting point to note here is that with constant transhybrid response the highest convergent constant was $= 1.1 \times 10^{-6}$; this means that the time-variance of the transhybrid response forces us to use a lower convergence constant which means slow time convergence.

Figure 32 and 33 shows the same output for the frequency domain algorithm. Figure 32 shows the near and far-end output with constant transhybrid response, Figure 33 shows the same outputs with time-variable transhybrid response. From these results we see that the frequency domain algorithm, because of its fast convergence, has almost the same performance for time variable transhybrid response. An important point to notice is that in both cases we used the same convergence constants (0.2 for the convergence constant and 0.9 for the power smoothing constant).

In summary, conversation of 50 sec length has been passed through the full duplex channel with time variant

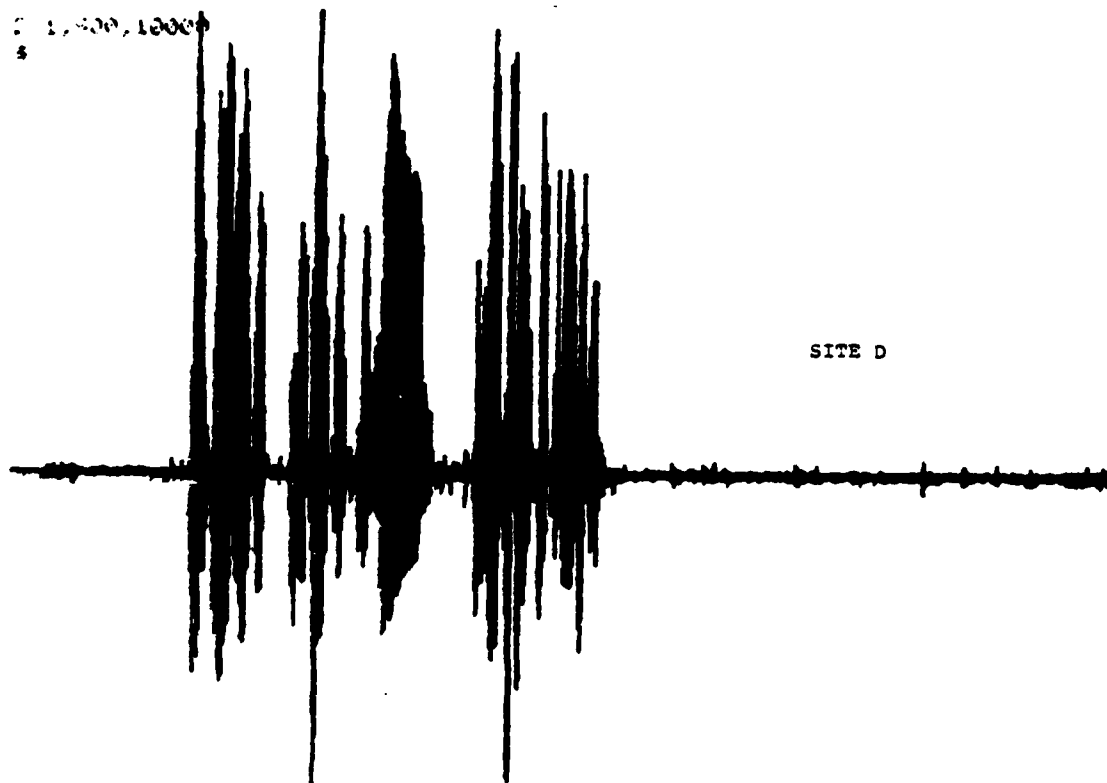
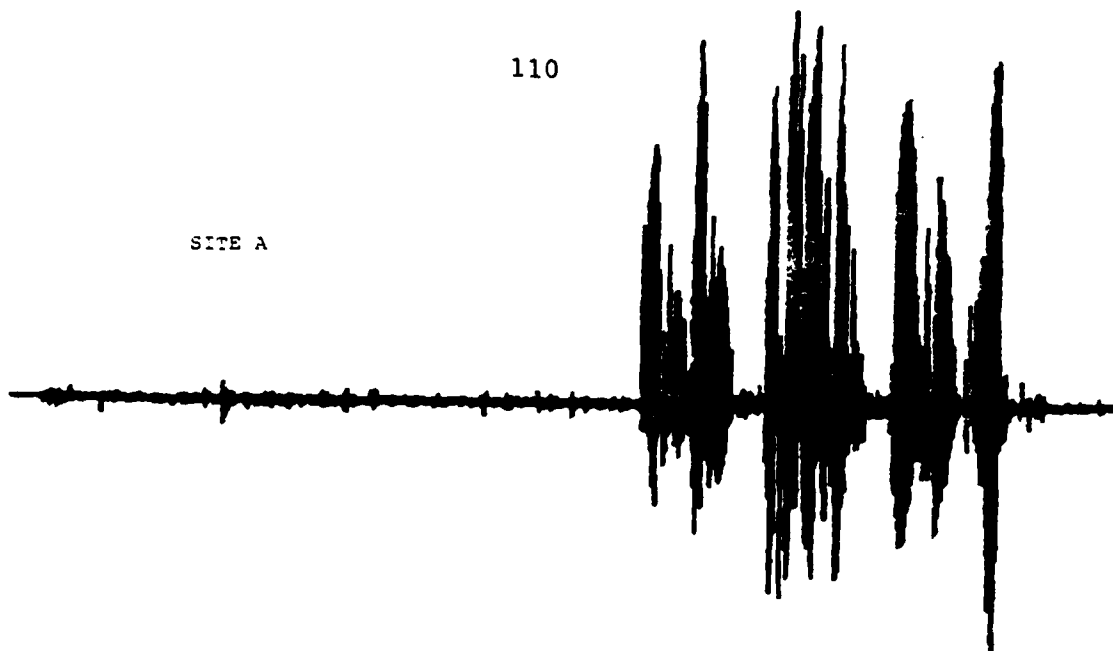
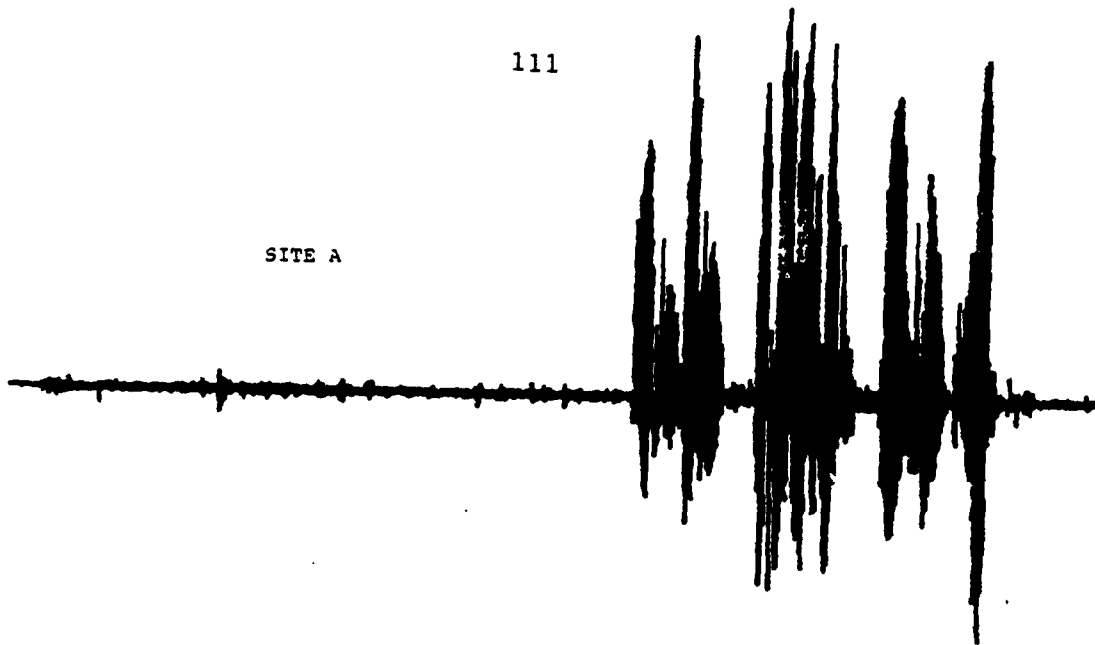


Fig. 32. Output at site A and D - Frequency domain with constant transhybrid response.

111

SITE A



SITE D

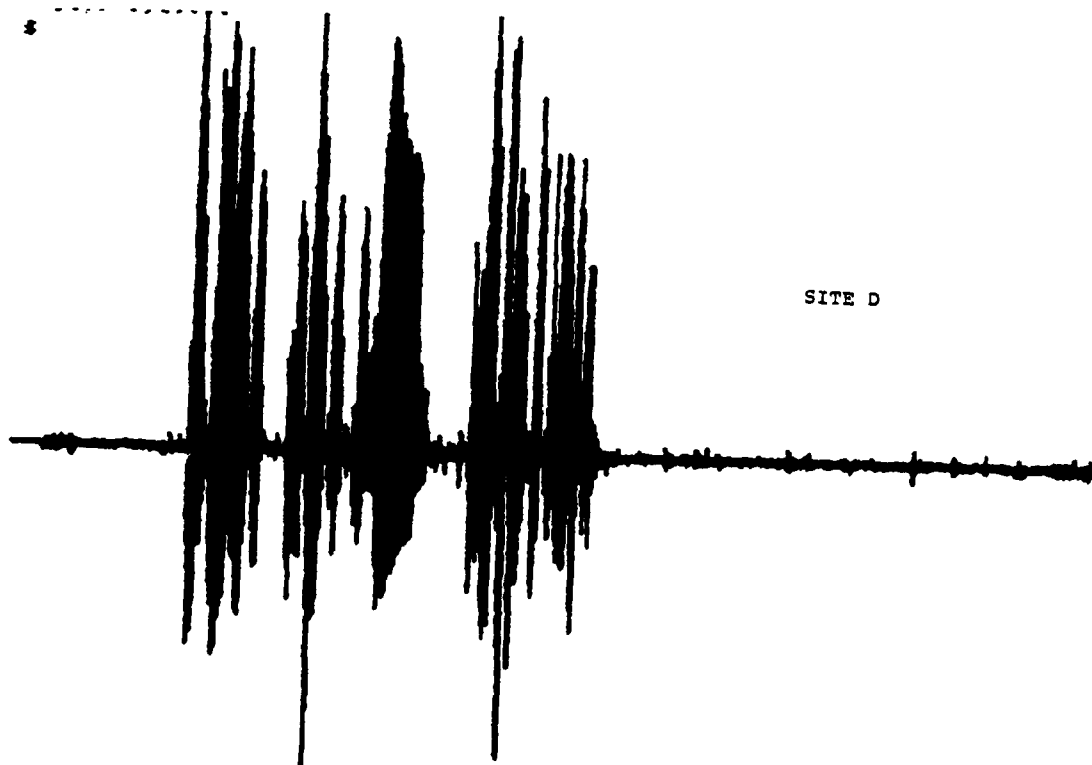


FIG. 33. Output at site A and D - Frequency domain with dynamic Transhybrid response. (Changes every 1 sec--display of 10 sec)

transhybrid response with both the LMS and the UFLMS algorithms. The test-bed simulation outputs at both ends of the full-duplex channel has been recorded. From listening to those outputs the superior behavior of the frequency domain algorithm is clear. This conversation contains a number of situations where both speakers talk simultaneously. In these situations we can observe the distortion introduced with the LMS algorithm compared to the good speech quality achieved by the UFLMS algorithm.

Summary

During this program a full-scale testbed simulation of the interfacing of a telephone to the wideband integrated network was completed. This simulation includes the use of different types of echo cancelling algorithms and an LPC vocoder, and allows for other studies to be carried on.

Echo cancelling algorithms were studied with a number of interesting conclusions. Because of the non-stationarity of the speech signals, made more so by the vocoder, standard LMS algorithms and lattice techniques are not adequate because of their convergence properties. With the vocoder in the loop, convergence must be faster than without the vocoder, because synthetic speech signals are not so rich in components as are the actual speech signals. An unconstrained frequency domain adaptive filter algorithm was the most effective.

The echo cancelling is limited by the nonlinearities of the system. It was found that a nonlinear adaptive filter could reduce the signal-to-noise ratio by a few more dB when used with a stationary system (the hybrid and line characteristics remain fixed). Further study would be needed before such a nonlinear adaptive filter should be used in a dynamic situation.

The possible use of an AGC in the system was studied,

and it was concluded that the AGC would cause more problems than it would help. The dynamic range problem of the analog signals is greatly reduced by improving the pitch and voicing detection algorithms.

The next step should be the breadboarding of a real time system to be tested in realistic situations which might show up problems not seen in a simulation.

8. References

- [1] D. G. Messerschmitt, "Considerations in the Evolution to Digital Telephony", manuscript provided to participants in "Digital Telephone" short course sponsored by U.C. Extension, Berkeley, CA, May 1978.
- [2] Steven B. Davis, John D. Markel, "Telephone Channel Equalization for Narrowband Speech Processors", Signal Technology, Inc., March 1, 1981.
- [3] M. M. Sondhi and D. Berkley, "Silencing Echos on the Telephone Network", Proc. of the IEEE, August 1980.
- [4] D. D. Parikh, S. B. Davis, J. D. Markel, "A Simplified Model of Echo in the Switched Telecommunications and Wideband Integrated Network", ARPA NSC NOTE No. 139. October 1979.
- [5] B. Widrow et al., "Adaptive Noise Cancelling, Principles and Applications", Proc. of the IEEE, Vol. 63, No. 12, December 1975.
- [6] D. L. Duttweiler, "A Twelve-channel Digital Echo Canceller", IEEE Trans. Commun., Vol-COM-20, pp. 647-653, 1978.
- [7] L. J. Griffiths, "An Adaptive Lattice Structure for Noise-Cancelling Applications", Conference Records 1978 ICASSP, pp. 387-90. Tulsa, OK, April 1978.
- [8] D. Mansour and A. H. Gray, Jr. "Unconstrained Frequency Domain Adaptive Filter", submitted to IEEE Trans. ASSP. October 1980.
- [9] M. G. Larimore, J. R. Treichler, and C. R. Johnson, Jr. "SHARF: An Algorithm for Adapting IIR Digital Filters". IEEE Trans. ASSP-28, No. 4, August 1980.
- [10] D. L. Dutweiller and Y. S. Chen, "A single-chip VLSI Echo Canceller", Bell System Technical Journal, February 1980.
- [11] B. H. Juang and J. D. Markel, "Cepstrally Based Pitch and Voicing Estimation with Statistical Assistance". ARPA NSC Note No. 140, October 1979.
- [12] Roy, R. J. and Sherman, J., "A Learning Technique for Volterra Series Representation", IEEE Trans. on Autom. Contr., Vol. AC-12, pp. 761-766, Dec. 6, 1967.

- [13] D. Mansour and A. H. Gray Jr., "System Identification of Hybrid Non-linearities", ARPA NSC Note No. 147, October 1981.
- [14] Volterra, V., "Theory of Functionals and Integrals and Integro-differential Equations", Dover, NY, 1957.
- [15] Schetzen, M., "A Theory of Non-linear Systems Identifications", International Journal of Control, Vol. 20, pp. 577-597, October 1979.
- [16] Eykhoff, P., "System Identification", Wiley, NY, 1979.
- [17] D. Parikh, S. C. Sinha, and N. Ahmed, "on a Modification of the SHARF Algorithm", in Proc. 22nd Midwest Symp. Circuits and Systems, Philadelphia, PA, June 1979.
- [18] Steven F. Boll, "Adaptive Noise Cancelling in Speech Using the Short-time Transform," ICASSP 80, Denver, CO, Vol. 3, pp. 692-695, April 1980.
- [19] M. Dentino, J. McCool and B. Widrow, "Adaptive filtering in the Frequency Domain," Proc. IEEE, Vol. 66, No. 17, pp. 1658-1659, December 1978.
- [20] N. J. Bershad and P. D. Feintuch, "Analysis of the Frequency Domain Adaptive Filter," Proc. IEEE, Vol. 67, No. 12, pp. 1658-1659, December 1979.
- [21] T. Waltzman and M. Schwartz, "Automatic Equalization Using the Discrete Frequency Domain," IEEE Trans. Info. Theory, Vol. IT-19, No. 1, pp. 59-68, January 1973.
- [22] E. R. Ferrara, "Fast Implementation of LMS Adaptive Filter," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-28, No. 4, August 1980.
- [23] M. Vidyasagar, "Nonlinear systems analysis", Prentice-Hall, Inc., Englewood Cliffs, NJ, 1978.
- [24] A.V. Oppenheim and R.W. Shaffer, Digital Signal Processing, Prentice Hall, NJ, 1975.
- [25] R.M. Gray, "On the Asymptotic Eigenvalue distribution of Toeplitz Matrices," IEEE Trans on Information Theory, Vol. IT-18, pp. 725-730, November 1972.
- [26] D. Parikh, D. Mansour, J. Markel, "Digital Simulation of Interfacing the Telephone Line to the Wideband system", ARPA NSC NOTE No. 144.

- [27] R.R. Bitmead and B.D.O. Anderson, "Lyapunov Techniques for the Exponential Stability of Linear Difference Equations with Random Coefficients," IEEE Trans. on Automat. Contr., Vol. AC-25, No. 4, August 1980.

9. Bibliography of Documents
Prepared Under N00014-78-C-0214

1. John D. Markel and Steven B. Davis
ARPA NSC Note No. 126
Preliminary Report on Telephone Input/Output for
Linear Prediction Processors
November 1, 1978
2. Steven B. Davis and John D. Markel
ARPA NSC Note 132
Experiments in Channel Equalization for Telephone
Input to LPC Systems
March 1, 1979
3. Steven B. Davis and John D. Markel
ARPA NSC Note 136
Additional Experiments in Telephone Channel Estimation
May 25, 1979
4. John D. Markel, Steven B. Davis, and Ted Applebaum
A Methodology for Studying Telephone Amplitude
Distortion Effects on Narrowband Speech Processors
IEEE International Conference ASSP
April, 1979
5. Steven B. Davis and John D. Markel
Experiments in Channel Equalization for Telephone
Input To LPC Systems
97th Meeting of the Acoustical Society of America
Massachusetts Institute of Technology
6. Steven B. Davis, Ted H. Applebaum and
John D. Markel
ARPA NSC Note No. 138
Telephone Channel Equalization Listening Tests
October 25, 1979
7. Steven B. Davis and John D. Markel
ARPA NSC Note 139
A Simplified Model of Echo in the Switched
Telecommunications and Wideband Integrated Networks
October 25, 1979
8. Biing-Hwang Juang, John D. Markel
ARPA NSC Note No. 140
Cepstrally Based Pitch and Voicing Estimation with
Statistical Assistance
October 20, 1979

9. Dak Parikh, David Mansour, and John Markel
 ARPA NSC Note 144
 Digital Simulation of Interfacing the Telephone Line
 to the Wideband System
10. Dak Parikh, David Mansour, A. H. Gray, Jr. and
 John D. Markel
 ARPA NSC Note 145
 Development and Study of Echo Cancelling Algorithms
 for Full Duplex Telephone Networks with Vocoders
 March 5, 1981
11. A. H. Gray, Jr., Biing-Hwang Juang, and David Mansour
 ARPA NSC Note 146
 Dynamic Range Observations In Wideband Satellite
 Communication
 September 5, 1981
12. David Mansour and A. H. Gray, Jr.
 ARPA NSC Note 147
 System Identification of Hybrid Nonlinearities
 October, 1981

10. Distribution List

Defense Documentation Center (12 copies)
Cameron Station
Alexandria, VA 22314

Office of Naval Research
Arlington, VA 22217

Information Systems Program (437) (2)
Code 200 (1)
Code 455 (1)
Code 458 (1)

Office of Naval Research (1)
Branch Office, Boston
495 Summer Street
Boston, MA 02210

Office of Naval Research (1)
Branch Office, Chicago
536 South Clark Street
Chicago, IL

Office of Naval Research (1)
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106

Office of Naval Research (1)
New York Area Office
715 Broadway - 5th Floor
New York, NY 10003

Naval Research Laboratory (6)
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. A. L. Slafkosky (1)
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380

Naval Ocean Systems Center (1)
Advanced Software Technology Division
Code 5200
San Diego, CA 92152

Mr. E. H. Gleissner (1)
 Naval Ship Research & Development Center
 Computation and Mathematics Department
 Bethesda, MD 20084

Captain Grace M. Hopper (008)(1)
 Naval Data Automation Command
 Washington Navy Yard
 Building 166
 Washington, D.C. 20374

Advanced Research Projects Agency (1)
 Information Processing Techniques
 1400 Wilson Boulevard
 Arlington, VA 22209

